

Regina de Cássia Nandi

**ISOMORFISMO DE GRAFOS APLICADO À COMPARAÇÃO DE
IMPRESSÕES DIGITAIS**

Curitiba

2006

Regina de Cássia Nandi

**ISOMORFISMO DE GRAFOS APLICADO À COMPARAÇÃO DE
IMPRESSÕES DIGITAIS**

**Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre. Programa de Pós-
Graduação em Informática, Setor de Ciências
Exatas, Universidade Federal do Paraná.**

Orientador: Prof. Dr. André Luiz Pires Guedes

Universidade Federal do Paraná

Curitiba

2006

"Uma longa viagem começa com um único passo".

Lao-Tsé

Dedico este trabalho à minha mãe Inês Netto Nandi, ao meu pai Pedro Bez Fontana Nandi e ao meu irmão Luís Eduardo Nandi, por todo apoio que tive durante o desenvolvimento deste trabalho.

Agradeço primeiramente à Deus pela oportunidade de cursar o Mestrado, pela força divina que me possibilitou concluir este trabalho e também por ter colocado no meu caminho muitas pessoas que de uma forma ou de outra me auxiliaram durante este tempo em que cursei o mestrado.

Agradeço à minha família e demais parentes que me apoiaram, torceram e pediram à Deus que eu tivesse força.

Agradeço ao meu orientador André Luiz Pires Guedes pela dedicação, paciência, oportunidade que me foi dada e todo o aprendizado que obtive durante a caminhada.

Agradeço à todos os amigos que fiz durante o tempo do mestrado e a amizade que me confiaram e que com alegria e possibilitaram superar os momentos difíceis pelos quais passamos.

Agradeço à minha amiga, Ledyvânia Franzotte, que desde nossa graduação sempre está do meu lado me apoiando e também à Francielle Biguelini pela amizade.

Agradeço também ao meu namorado, Lucas Giovane Zanette, pela paciência, amor e compreensão demonstrado mesmo quando estávamos muito longe.

SUMÁRIO

RESUMO.....	iii
ABSTRACT.....	iv
LISTA DE TABELAS.....	v
LISTA DE FIGURAS	vi
LISTA DE FIGURAS	vi
1. INTRODUÇÃO	1
1.1 <i>Motivação.....</i>	<i>2</i>
1.2 <i>Objetivos</i>	<i>2</i>
1.3 <i>Organização do trabalho.....</i>	<i>3</i>
2. CONCEITOS DE GRAFOS.....	4
2.1 <i>Definições gerais</i>	<i>4</i>
2.2 <i>Isomorfismo de Grafos.....</i>	<i>6</i>
3. CONCEITOS DE BIOMETRIA E IMPRESSÕES DIGITAIS.....	8
3.1 <i>Biometria.....</i>	<i>8</i>
3.2 <i>Impressões digitais</i>	<i>9</i>
3.2.1 <i>Critérios para análise de impressões digitais.....</i>	<i>10</i>
3.3 <i>Sistemas automáticos de identificação de impressão digital.....</i>	<i>11</i>
4. ISOMORFISMO DE GRAFOS APLICADO À COMPARAÇÃO DE IMPRESSÕES DIGITAIS	13
4.1 <i>O problema.....</i>	<i>13</i>
4.2 <i>Metodologia adotada para encontrar a solução.....</i>	<i>15</i>
4.3 <i>Resolução do problema.....</i>	<i>16</i>
4.3.1 <i>Encontrar minúcias nas imagens de impressão digital</i>	<i>16</i>
4.3.2 <i>Geração do grafo.....</i>	<i>17</i>
4.3.3 <i>Cálculo do isomorfismo</i>	<i>17</i>
4.4 <i>Portugol do algoritmo.....</i>	<i>18</i>
4.5 <i>Características necessárias para os grafos G_C e G_F.....</i>	<i>22</i>
4.6 <i>Análise do Algoritmo</i>	<i>24</i>
4.7 <i>Otimização proposta e implementada: isomorfismo com controle no tamanho das arestas</i>	<i>25</i>
4.8 <i>Estado da arte - trabalhos semelhantes.....</i>	<i>27</i>

5. Desenvolvimento	29
5.1 Linguagem.....	29
5.2 Implementação	29
5.3 A ferramenta de testes.....	31
6. EXPERIMENTOS REALIZADOS E RESULTADOS ALCANÇADOS.....	37
6.1 Experimentos.....	37
6.1.1 Experimento nº 1: Fragmento e Candidata sendo representados por grafo Estrela de centro aleatório	38
6.1.1.1 Experimento nº 1a: sem controle no tamanho das arestas - discussão do Resultado	39
6.1.1.2 Experimento nº 1b: com controle no tamanho das arestas - discussão do Resultado	39
6.1.2 Experimento nº 2: Fragmento e Candidata sendo representadas por grafos Bipartite	40
6.1.2.1 Experimento nº 2a: sem controle no tamanho das arestas - discussão do Resultado	41
6.1.2.2 Experimento nº 2b: com controle no tamanho das arestas - discussão do Resultado	43
6.1.3 Experimento nº 3: Fragmento e Candidata sendo representadas por grafos Completos	44
6.1.3.1 Experimento nº 3a: sem controle no tamanho das arestas - discussão do Resultado	45
6.1.3.2 Experimento nº 3b: com controle no tamanho das arestas - discussão do Resultado	46
6.1.4 Experimento nº 4: Fragmento representada por Grafo Estrela e Candidata representado por Grafo Completo	47
6.1.5 Experimento nº 5: Fragmento representada por Grafo Estrela (com poucos vértices) e Candidata representada por Grafo Completo	50
7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	52
7.1 Trabalhos futuros	52
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	53
ANEXO A – ENTRADAS PARA O EXPERIMENTO Nº 1	56
ANEXO B – ENTRADAS PARA O EXPERIMENTO Nº 2.....	57

RESUMO

Este trabalho apresenta uma metodologia para comparação de impressões digitais baseada em isomorfismo de grafos. Esta metodologia é baseada na construção de grafos a partir das minúcias encontradas na impressão digital e retorna verdadeira quando a árvore geradora de um grafo G_F , que representa uma impressão digital a ser comparada é isomorfa a um subgrafo de G_C , que representa a impressão digital candidata para ser o par da primeira. Cada vértice do grafo representa uma minúcia encontrada na imagem da impressão digital e é rotulado com o seu tipo e posição geométrica na imagem. As arestas representam uma relação de vizinhança entre as minúcias. A comparação é realizada através do cômputo do isomorfismo dos grafos. Pretendeu-se com esta metodologia ser flexível, evitando problemas que podem acontecer quando há minúcias faltando em uma das impressões digitais e diferenças geométricas tais como: rotações, translações e escalas das imagens das impressões digitais. O algoritmo foi implementado e testado usando imagens adquiridas da Internet para as quais as minúcias foram encontradas em um processo manual. Os testes foram realizados com diferentes tipos de grafos construídos a partir das minúcias das impressões digitais, definindo diferentes relações de vizinhança entre elas. Neste trabalho também é apresentada uma abordagem elaborada para o cálculo do isomorfismo com consideração no tamanho das arestas com a finalidade de reduzir o número de comparações realizadas.

ABSTRACT

This work presents a methodology to fingerprint matching using graph isomorphism. This methodology is based on the graph construction from minutiae found in fingerprints and returns true when the spanning tree of a graph G_F (that represents a fingerprint to be compared) is isomorph to a subgraph of G_C (that represents the candidate fingerprint to be the pair of the first one). Each vertex represents one minutiae found in the fingerprint image and is labelled with its type and geometric position in the image. The edges represent a relation of neighborhood between these features. The matching is done through the computation of graph isomorphism. With this methodology, it was intended to be flexible, avoiding problems that can happen when there are no minutiae enough in some places of the fingerprint and geometric problems such as: rotation, translation and scale of fingerprint images. The algorithm was implemented and tested using images from the Internet which the minutiae had been found using a manual process. The tests has been done with different types of graphs constructed from the minutiae of the fingerprints, defining different relations between them. Also, in this work is elaborated a new method to calculate the isomorphism, considering the size of the edges to reduce the number of comparisons done.

LISTA DE TABELAS

Tabela 1 - Layout do arquivo com minúcias pré-detectadas	32
Tabela 2 - Resumo da Comparação	35
Tabela 3 – Dados de entrada para os experimentos 1, 2 e 3.....	37
Tabela 4 – Resultado principal do experimento nº 2a.....	41
Tabela 5 - Resultado principal do experimento nº 2b	43
Tabela 6 - Resultado principal do experimento nº 2a	45
Tabela 7 - Resultado principal do experimento nº 2b	46
Tabela 8 - Resultados principais do experimento nº 4	48
Tabela 9 – Mapeamentos encontrados para o experimento 5.....	51

LISTA DE FIGURAS

Figura 1 – Alguns tipos de minúcias.....	1
Figura 2 – Grafo G	4
Figura 3 – Um grafo G e seus subgrafos	5
Figura 4 – Grafos isomorfos	6
Figura 5 - Classificação global das impressões digitais segundo HENRY.....	10
Figura 6 – Resumo da Metodologia proposta.....	16
Figura 7 – Portugal: núcleo da resolução do problema.....	18
Figura 8 – Portugal: continuação do teste de isomorfismo	19
Figura 9 – Exemplo de execução do algoritmo	21
Figura 10 – Diagrama de classes da implementação realizada.....	31
Figura 11 – <i>Snapshot</i> da ferramenta utilizada para realizar os experimentos	32
Figura 12 – Grafos criados para as impressões digitais	33
Figura 13 – Resultado da comparação com vértices que foram ‘casados’ em destaque.....	34
Figura 14 – Arquivo de log mantido durante o cálculo do isomorfismo	36
Figura 15 – Grafos de Entrada para o Experimento nº 1.....	38
Figura 16 – Grafos de Entrada para o Experimento nº 2.....	41
Figura 17 – Relação de isomorfismo entre os vértices de G_F e G_C	42
Figura 18 – Grafos de Entrada para o Experimento nº 3.....	45
Figura 19 – Grafos de Entrada para o Experimento nº 4.....	47
Figura 20 – Relação de isomorfismo entre os vértices de G_F e G_C	49
Figura 21 – Grafos de Entrada para o Experimento nº 5.....	50
Figura 22 – Visualização de alguns dos mapeamentos encontrados para o experimento nº 5.	51

1. INTRODUÇÃO

A análise de impressões digitais é uma técnica biométrica que tem sido utilizada há muito tempo com o objetivo de identificação de pessoas. Os egípcios antigos e os chineses já as utilizavam na identificação de criminosos e no registro de transações comerciais [MAT1993]. A imutabilidade e a individualidade das impressões digitais são dois dos principais motivos para sua utilização na identificação civil e criminal de pessoas. A imutabilidade se refere ao caráter permanente do padrão da impressão digital e a individualidade se caracteriza pela unicidade dos detalhes das linhas que a constituem e a diferenciam de outras impressões digitais.

As minúcias (Figura 1) referem-se aos mínimos e pequenos detalhes presentes em uma impressão digital.



Figura 1 – Alguns tipos de minúcias

Fonte: [ABE, 2002]

A existência de um sistema de identificação de pessoas através de impressões digitais exige um método de comparação das características encontradas nas imagens de impressões digitais adquiridas. Qualquer que seja o método utilizado – manual ou automático – deve ter como entrada características obtidas a partir de um processo prévio de análise sobre as imagens das impressões digitais. Esta comparação visa analisar as características de duas imagens informando se foram adquiridas do mesmo dedo ou não.

A Teoria dos Grafos tem sido utilizada em vários ramos da ciência pela sua capacidade de representar entidades e suas relações. Em reconhecimento de padrões, a utilização de grafos é bastante explorada onde um determinado padrão é analisado e

características desejadas são encontradas, e então podem ser construídos grafos que representem de alguma forma a descrição do padrão, regiões de adjacência, ou qualquer relação mensurável entre um conjunto de pontos [MAT1993].

Este trabalho apresenta uma metodologia para a comparação de impressões digitais. Esta metodologia propõe que a partir de características obtidas de duas imagens de impressões digitais sejam construídos grafos para representá-las. A comparação das impressões propriamente dita é realizada através do cômputo do isomorfismo dos grafos construídos.

1.1 Motivação

Podemos citar algumas questões motivadoras para representar as impressões digitais como grafos e aplicar o isomorfismo de grafos na em sua comparação:

- A fase de comparação de impressões digitais é um processo de grande importância em um sistema de reconhecimento de impressões digitais e a pesquisa sobre a aplicação do uso de grafos nesta fase visa contribuir com mais uma experiência para realizar esta atividade;
- Desenvolvimento de um algoritmo para comparação baseado em grafos que possua resultados satisfatórios em termos de desempenho e resultados alcançados;
- Uma vez que o isomorfismo de subgrafos pode ser utilizado para analisar a disposição de objetos numa cena, auxiliando na detecção de um objeto embutido em um padrão, pretende-se verificar e validar a utilização do isomorfismo em mais um experimento nesta área de visão computacional.

1.2 Objetivos

O objetivo principal deste trabalho é explorar o uso de grafos na comparação de impressões digitais. Imagens de duas impressões digitais são representadas através de grafos construídos a partir das minúcias pré-encontradas em cada impressão digital. As minúcias são características encontradas nas impressões digitais, que, para este trabalho estão divididas em 2 (dois) tipos: final de linha e bifurcação.

Cada vértice do grafo construído representa uma minúcia encontrada na imagem da impressão digital e é rotulado com o seu tipo e posição geométrica e as arestas representam

uma relação de vizinhança entre estas características encontradas. A comparação propriamente dita é realizada através do cômputo do isomorfismo sobre os grafos gerados. De forma semelhante ao que acontece na realidade, as imagens de impressão digital utilizadas no presente trabalho para a comparação nem sempre são bem comportadas, podendo possuir transformações como escala, translação e rotação. Outro caso comum, também simulado aqui, é que em um sistema de reconhecimento de impressões digitais tem-se previamente uma imagem de qualidade armazenada no banco de dados, e no momento da comparação tem-se somente um fragmento da impressão na imagem adquirida. Este fragmento tende a ter uma menor quantidade de minúcias encontradas do que a imagem que foi previamente armazenada no banco de dados, no momento do cadastro de uma impressão digital.

Os testes foram feitos através da variação do tipo de grafo utilizado para representar a impressão digital, onde se pretendeu encontrar o tipo mais adequado para resolver o problema da comparação de impressões digitais.

1.3 Organização do trabalho

Este texto apresenta um embasamento teórico, o desenvolvimento deste trabalho, os experimentos e os resultados alcançados. Está dividido em 7 (sete) capítulos. No capítulo 2 e 3, conceitos sobre grafos e impressões digitais são explorados. No capítulo 4 é abordado o tema deste trabalho, onde são explanados os problemas e apresentada a metodologia proposta para comparar as impressões através do isomorfismo de grafos e, posteriormente, detalhes sobre a implementação são apresentados no capítulo 5.

No capítulo 6, os experimentos e resultados alcançados são apresentados. Por fim, o capítulo 7 apresenta as considerações finais e as sugestões de trabalhos futuros.

2. CONCEITOS DE GRAFOS

Este capítulo contém definições sobre grafos, subgrafos e isomorfismo de grafos para que o leitor possa compreender de uma melhor forma o desenvolvimento deste trabalho.

2.1 Definições gerais

Os grafos são abstrações matemáticas que representam situações reais através de pontos e linhas, onde os pontos apresentam alguma informação e as linhas indicam o relacionamento entre dois pontos quaisquer. Os grafos podem ser aplicados na solução de problemas da genética, química, telecomunicações, computação, engenharia elétrica, entre outros [GRE2005].

Um grafo G é um par ordenado (V, E) tal que E (conjunto de arestas) é um subconjunto do conjunto dos pares não ordenados de V (conjunto não vazio de vértices). Na resolução de algum problema geralmente considera-se somente grafos finitos em que V e E são sempre finitos. Uma aresta $\{x,y\}$ liga os vértices x e y e é denotada por xy onde x e y são os vértices finais desta aresta. Se $xy \in E$ então x e y são adjacentes ou vértices vizinhos de G e são incidentes à aresta xy . Duas arestas são adjacentes se elas tem exatamente um vértice em comum [BOL1979]. Um exemplo de grafo pode ser conferido na Figura 2:

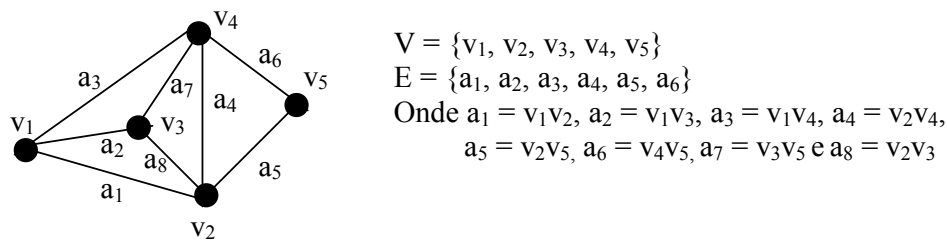


Figura 2 – Grafo G

Um grafo $G'=(V',E')$ é subgrafo de um grafo $G=(V,E)$ se $V' \subseteq V$ e $E' \subseteq E$. Neste caso, pode-se escrever $G' \subseteq G$. Com base nesta definição, os seguintes enunciados podem ser verificados [GRE2005; GAG2005]:

- Todo grafo é subgrafo dele mesmo.
- O subgrafo de um subgrafo de G é um subgrafo de G .
- Um vértice de G é um subgrafo de G .
- Uma aresta de G com os dois vértices que ela liga é um subgrafo de G .

Se G' contém todas as arestas de G que ligam dois vértices em V' então G' é denominado subgrafo induzido. Se G' contém todos os vértices de G mas não necessariamente todas as arestas, então G' é denominado subgrafo gerador [BOL1998].

A Figura 3 ilustra o grafo G e quatro de seus subgrafos, A, B, C e D.

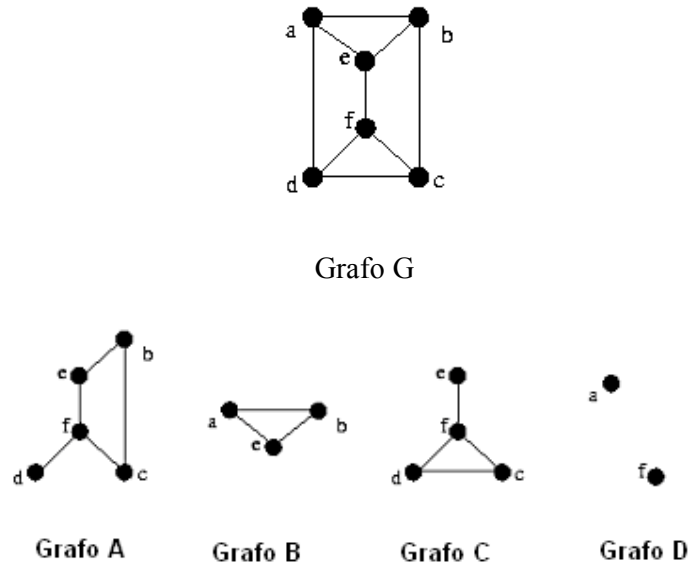


Figura 3 – Um grafo G e seus subgrafos

A ordem de um grafo G é o número de vértices em G ; e é denotada por $|V|$. O tamanho de G é o número de arestas em G e é denotado por $|E|$. $G(n,m)$ denota um grafo arbitrário de ordem n e tamanho m . O tamanho de um grafo de ordem n é no mínimo 0 e no máximo $n(n-1)/2$. Um grafo de ordem n e tamanho $n(n-1)/2$ (correspondendo a todas as possíveis escolhas de pares de vértices) é chamado grafo completo e é denotado por K_n ; um grafo vazio E_n tem ordem n e nenhuma aresta. Em K_n cada dois vértices são adjacentes, enquanto em E_n , nenhum par de vértices são adjacentes. O grafo $K_1 = E_1$ é denominado trivial [BOL1998].

2.2 Isomorfismo de Grafos

Em se tratando de comparação de grafos, o isomorfismo de grafos é um problema de fundamental importância, mas de difícil tratamento. Atualmente nenhum algoritmo determinístico de complexidade de tempo polinomial é conhecido para determinar quando dois dados grafos G_F e G_C são isomorfos. Um algoritmo determinístico eficiente garante uma solução em tempo $T(n)$, onde $T(n)$ é um polinômio em n , a ordem do grafo. Um problema intimamente ligado, porém inerentemente mais complexo é o isomorfismo de subgrafos no qual dado dois grafos finitos, G_F e G_C , determinar se G_F é isomorfo à G_C ou a um subgrafo de G_C . Problemas de isomorfismo de grafos e/ou subgrafos estão presentes em diversos campos de pesquisa, tais como: química, recuperação de informação, lógica e teoria de redes [COR1970].

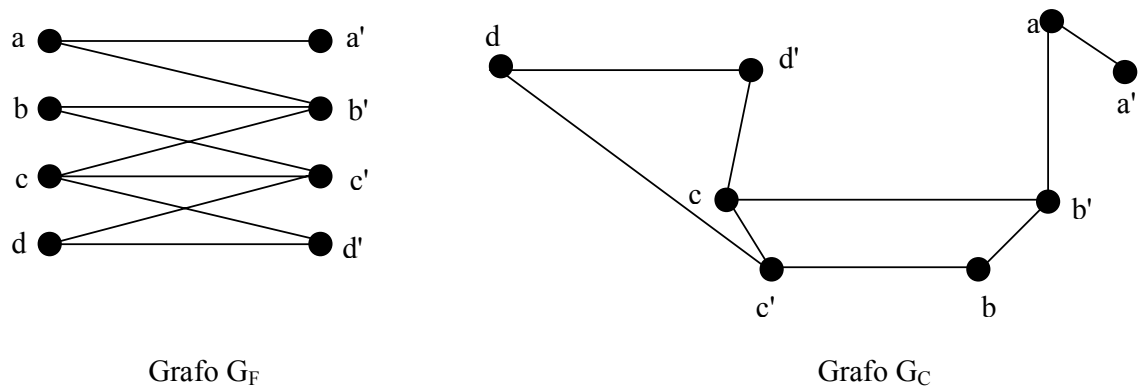


Figura 4 – Grafos isomorfos

O grafo G_F (Figura 4) pode ser desenhado de diferentes formas, sendo uma delas o grafo G_C . Estes dois grafos são denominados isomorfos [GRE2005; OLI2005].

Formalmente, $G_F=(V_F, E_F)$ e $G_C=(V_C, E_C)$ são isomorfos se satisfazem a seguinte condição [BOL1998; ATL2006]:

- Existe uma função bijetora $f: V_F \rightarrow V_C$ tal que $vw \in E_F$ se e somente se $f(v)f(w) \in E_C$.

A nomenclatura $G_F \cong G_C$ indica que um grafo G_F é isomorfo à outro grafo G_C .

Como consequências do isomorfismo, têm-se:

- $|G_F| = |G_C|$ e $e(G_F) = e(G_C)$, ou seja, possuem a mesma ordem (número de vértices) e tamanho (número de arestas).
- G_F e G_C , tem o mesmo vetor de grau, que é a sequência dos graus (ou ordens) dos vértices de cada grafo, colocada em ordem não crescente.

O problema do isomorfismo de grafos, pertence à classe NP. Não se sabe, porém, se este é polinomialmente solúvel, ou se pertence à família dos problemas de complexidade mais alta desta classe, os NP-completos. A hipótese mais aceita, é que esteja numa região entre a classe P e a dos problemas NP-completos [OLI2005].

3. CONCEITOS DE BIOMETRIA E IMPRESSÕES DIGITAIS

O objetivo deste capítulo é fornecer um embasamento teórico sobre biometria e como funciona a identificação de pessoas através da análise de impressões digitais.

3.1 Biometria

A biometria é um recurso que permite identificar uma pessoa com base em alguma informação conhecida como suas características físicas ou comportamentais. Este método de identificação é preferível sobre os métodos tradicionais envolvendo senhas e números de identificação pessoal por duas razões principais: a pessoa a ser identificada necessita estar fisicamente presente no local da identificação e a identificação baseada em técnicas biométricas não requer que uma pessoa carregue consigo uma senha, uma vez que a identificação está presente em cada pessoa.

Um sistema computacional baseado em biometria é essencialmente um sistema de reconhecimento de padrões, onde a identificação pessoal é feita através da comparação de uma característica específica (fisiológica ou comportamental) possuída por uma pessoa, característica esta que deve ter um modelo previamente armazenado no sistema que será comparada com o novo modelo que será fornecido pela pessoa quando ela requerer sua identificação.

Atualmente, várias características biométricas já estão sendo utilizadas em sistemas computacionais para identificação de pessoas em tempo real e dentre as mais populares podemos citar: face, íris e comparação de impressões digitais. Entretanto, existem outros sistemas biométricos que utilizam a varredura da retina, voz, assinatura, geometria da mão e análise do ritmo de escrita para identificação de pessoas. A identificação de uma pessoa por meio de suas características biométricas possui as seguintes vantagens [FRA2002]: i) são dados presentes na maioria das pessoas; ii) estão sempre disponíveis; iii) não podem ser transferidas para outra pessoa; iv) não podem ser esquecidas ou roubadas e v) sempre variam de pessoa para pessoa.

Uma importante questão em se tratando de desenvolver um sistema deste tipo é determinar a forma como a identificação acontece. Dependendo do contexto, um sistema biométrico pode ser operado tanto em um modo de verificação, no qual o sistema tenta responder à pergunta do usuário: “Sou eu quem eu digo ser?” onde um modelo registrado no

banco de dados é comparado com um outro padrão, ou no modo de identificação, no qual o sistema tenta responder: “Quem sou eu?”, onde verifica se um dado padrão existe no banco de dados, analisando todo o banco de dados [ARI2005].

Uma limitação dos sistemas de identificação baseados em biometria é que sua precisão nem sempre é satisfatória sendo que estes sistemas algumas vezes aceitam um impostor e/ou rejeitam um usuário autêntico. Esses dois erros dependem do ponto limiar de operação do sistema (*threshold*) [JAI2001]. Assim, o desempenho de um sistema biométrico é medido através da quantidade de erros que comete. Os erros são medidos de duas maneiras: a) pela quantidade de pessoas com permissão que são barradas (falso negativo) e b) pela quantidade de pessoas sem permissão que são deixadas passar (falso positivo).

3.2 Impressões digitais

Entre todas as técnicas biométricas de identificação, o reconhecimento das impressões digitais é o método mais antigo (há muito tempo é utilizado para reconhecer criminosos), que tem sido utilizado com sucesso em muitas aplicações, principalmente pelo fato de ser uma das técnicas mais usadas e comprovadas, além de ser uma forma de verificação não muito intrusiva, amigavelmente-usável e que relativamente não tem custo elevado se comparado a outras técnicas de identificação biométrica.

“Uma impressão digital é caracterizada por um padrão de fluxo suave de vales e linhas (cristas) alternantes paralelas em muitas regiões da ponta dos dedos. Outras características permanentes e não-permanentes como: cicatrizes, cortes, contusões e rachaduras podem estar presentes em uma impressão digital” [JAI2002].

A datiloscopia (ciência que trata da identificação de uma pessoa por meio das impressões digitais), baseia-se em quatro princípios científicos, que são requisitos padrões para o reconhecimento de uma pessoa através de sua característica biométrica:

- Perenidade: as impressões digitais se formam no sexto mês de vida intra-uterina e se mantém inalteradas durante toda a vida de uma pessoa;
- Imutabilidade: os desenhos digitais não mudam sua forma original em nenhum estágio da vida;
- Classificabilidade: é a possibilidade de classificarmos as impressões digitais em categorias dependendo de como é o desenho encontrado em uma impressão digital;

- Variabilidade: Propriedade dos desenhos digitais em variar de dedo para dedo e de pessoa para pessoa.

3.2.1 Critérios para análise de impressões digitais

A unicidade das impressões digitais das pessoas ou até mesmo de dedos diferentes de uma mesma pessoa pode ser comprovada sob dois critérios principais. O primeiro critério está relacionado ao grupo da impressão (classificação global) e o segundo critério está relacionado à distribuição planar das minúcias (classificação local). Em (HENRY, 1905) *apud* [COS2001] uma impressão digital pode ser classificada globalmente em (Figura 5):

- Arco Plano: não apresenta delta (ponto de divergência das linhas que compõem a impressão digital que se apresenta na forma de um pequeno triângulo); as linhas que compõem a impressão digital formam-se em um lado e tendem a sair pelo outro lado;
- Arco Angular: apresentam acentuada elevação das linhas no centro da impressão digital. Pode apresentar um delta, mas sem linha entre o delta e o núcleo (centro da impressão digital);
- Presilha Interna: as linhas formam-se à esquerda do observador, curvam-se, e tendem a voltar para o mesmo lado;
- Presilha Externa: as linhas formam-se na direita do observador, curvam-se no centro da impressão e tendem a voltar para o mesmo lado;
- Verticilo: apresentam dois deltas, sendo um à direita e outro à esquerda do observador.

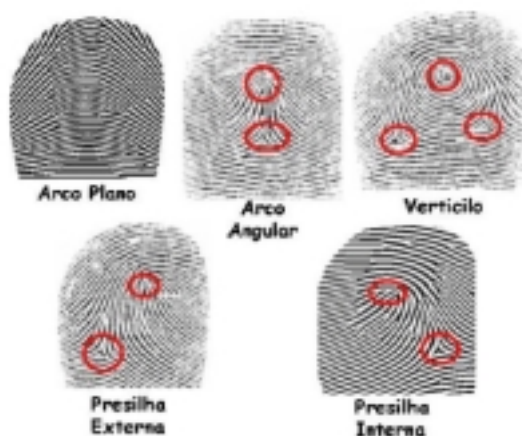


Figura 5 - Classificação global das impressões digitais segundo HENRY.

Fonte: [COS2001] - Deltas e núcleos em destaque.

Resumidamente, o critério de grupo é definido pelo comportamento das linhas nos dedos e também pela presença ou não, bem como a distribuição de duas características adicionais: o núcleo (centro da impressão digital) e o delta (região triangular encontrada na impressão digital). Esta visão macroscópica permite, num processo de comparação de duas imagens, afirmar a não identidade no caso de grupos globais diferentes. Quando os grupos globais de duas impressões digitais são iguais, o critério mais específico, das minúcias, permite verificar a identidade de uma pessoa em função da ocorrência ou não de um mapeamento posicional entre os diversos tipos de minúcias existentes em uma impressão digital. [TEC2005].

Segundo um levantamento feito pelo FBI (*Federal Bureau of Investigation* - Polícia Federal Americana) cerca de oitenta tipos de características (minúcias) podem ser encontradas nas linhas (cristas) das impressões digitais. Porém os finais de linha e bifurcações são os aspectos extraídos para a verificação, pois são freqüentemente encontradas numa análise [COS2001].

A técnica geralmente utilizada para realizar a comparação de impressões digitais, analisa as minúcias encontradas e a sua distribuição na impressão digital. Uma representação baseada em minúcias inclui a localização e orientação das mesmas, podendo incluir um ou mais atributos como orientação do dedo, localizações do centro ou delta, e classe da impressão digital [JAI2001].

3.3 Sistemas automáticos de identificação de impressão digital

Os esquemas populares de representação de impressões digitais desenvolveram-se a partir de um sistema intuitivo projetado por peritos em impressões digitais que visualmente comparam as impressões digitais. Entre as vantagens da identificação de uma pessoa por meio das suas impressões digitais pode-se destacar a rapidez e a confiança, aliada ao baixo preço e o pequeno tamanho dos leitores óticos.

Os sistemas computacionais de identificação biométrica que comparam impressões digitais são associados à sigla AFIS, e são compostos basicamente por quatro fases distintas [BOL1997]:

- Aquisição dos dados: a captura da impressão digital é efetuada através de um dispositivo (sensor) que transforma os traços que definem a impressão digital de um dedo em uma imagem digital;

- Representação – nesta fase, um modelo de como a imagem da impressão digital, ou a impressão digital propriamente dita deverá ser armazenado no sistema. A escolha de uma representação para o dado constitui a essência da verificação da impressão digital, e tem implicações no desenvolvimento do resto do sistema;
- Extração das características – é nesta fase que a imagem é analisada, buscando por dados que sejam capazes de representarem esta impressão digital e que também sejam a informação necessária para diferenciá-la das demais. Porém isso não é uma tarefa trivial, necessitando-se muitas vezes aplicar algoritmos para eliminar ou corrigir pequenos ruídos e falhas que podem acontecer na imagem durante a fase de captura. Se um algoritmo eficiente para extração das características não for aplicado, algumas características poderão não ser detectadas ou falsas características podem ser detectados na impressão digital;
- Verificação ou Comparação – fase final, onde acontece a comparação de uma impressão digital com o modelo previamente armazenado no sistema para identificar ou não uma pessoa. Embora a fase de comparação de duas imagens seja inspirada no procedimento manual (comparação baseada nas minúcias), existe um grande número de abordagens para este passo, tais como [JAI2003]:
 - comparação baseada na correlação, onde duas imagens de impressões digitais são sobrepostas e a correlação entre os *pixels* é computada para calcular o alinhamento, comparando principalmente os modelos globais das linhas;
 - comparação baseada nas minúcias, que consiste essencialmente em encontrar o alinhamento das minúcias extraídas das imagens a serem comparadas, verificando os pares alinhados;
 - comparação baseada nas características das linhas, pelo motivo da extração das minúcias ser muito difícil em imagens de baixa qualidade, este item aborda a comparação das impressões em termos das características extraídas a partir do modelo das linhas.

A comparação tipicamente produz uma pontuação quantificando a similaridade entre as duas representações. Se esta pontuação está dentro de um certo limite de segurança as representações são classificadas como sendo originadas a partir da mesma imagem da impressão digital.

4. ISOMORFISMO DE GRAFOS APLICADO À COMPARAÇÃO DE IMPRESSÕES DIGITAIS

Este capítulo descreve a metodologia utilizada para resolver o problema da comparação de impressões digitais. Esta metodologia proposta e desenvolvida neste trabalho é baseada no isomorfismo de grafos que são construídos para representar as minúcias do tipo bifurcação e final de linha previamente localizadas nas imagens das impressões digitais.

4.1 O problema

O objetivo principal deste trabalho é comparar imagens de impressões digitais que são representadas por grafos, utilizando isomorfismo de grafos. Para verificar a acurácia da metodologia proposta para comparação das impressões digitais - aqui denominadas por ID's - tem-se previamente dados de duas imagens denominadas neste trabalho por Fragmento e Candidata. Estes dados são adquiridos a partir de um processo manual, onde as minúcias do tipo final de linha e bifurcação bem como suas posições na imagem são encontradas e armazenadas separadamente em um arquivo texto para serem lidas e contadas como dados detectados nas impressões digitais no processo de comparação. Desta forma, pretende-se verificar por meio da construção de grafos constituídos a partir destas minúcias demarcadas, se a imagem Fragmento e Candidata possivelmente foram adquiridas a partir do mesmo dedo e conseqüentemente, da mesma pessoa.

Optou-se por denominar Candidata uma imagem de qualidade, similar a que é armazenada num banco de dados e confrontada com a outra imagem, e Fragmento, que é similar às imagens que são geralmente adquiridas no momento da comparação, ou seja, com uma quantidade razoavelmente menor de minúcias que a impressão Candidata. Isto se deve ao fato que, em um sistema de reconhecimento de impressões digitais, a imagem Fragmento, freqüentemente é incompleta, ou seja, é uma imagem com poucas minúcias.

O processo de comparação de impressões digitais basicamente pode ser realizado em 3 níveis:

- 1) comparação no nível de impressões digitais, no qual todos os dados presentes em uma impressão digital são analisados (minúcias e linhas que as compõem). Geralmente é feito manualmente/visualmente e é um processo bem trabalhoso.

- 2) comparação no nível de conjunto de pontos, no qual pontos específicos que representam características encontradas nas impressões são analisados. Pode ser realizado tanto manualmente como computacionalmente.
- 3) comparação no nível de grafos, onde a partir de um conjunto de pontos, um grafo é criado (seus vértices representam estes pontos), e as impressões digitais são comparadas através da análise de seus grafos. Pode ser realizado tanto manualmente como computacionalmente.

O presente trabalho pretendeu-se comparar impressões digitais no terceiro nível, ou seja, a partir do conjunto de pontos (nível 2), um grafo é criado para cada impressão e a análise é feita sobre os seus grafos que estabelecem uma pré-determinada relação de vizinhança entre os pontos.

Deve-se lembrar que a análise no terceiro ou no segundo nível pode nos dar uma resposta do tipo “sim”, mas esta resposta não implica necessariamente que as impressões digitais, cujos grafos ou conjunto de pontos que foram criados representem necessariamente a mesma impressão digital. Isso acontece porque apesar dos grafos ou pontos se encaixarem, pode ser que o número de pontos analisados seja muito pequeno e que realmente este número faça com que existam falsos positivos.

A análise do ponto de vista do nível 3 através dos grafos é uma análise melhor que a do nível 2, uma vez que o grafo com suas arestas definidas de forma adequada, pode fornecer um encaixe mais limitado. Por outro lado, se um cuidado com o tipo de grafo escolhido não for tomado, as arestas podem atrapalhar na comparação. Este detalhe será discutido com detalhes na seção 4.5 que trata das características a serem garantidas para os tipos de grafos escolhidos.

O problema que pretendeu-se resolver, está definido matematicamente na seqüência.

Tendo um conjunto de pontos X (minúcias encontradas em uma ID_F , denominada aqui por Fragmento) e um conjunto de pontos S (minúcias encontradas em uma ID_C , denominada Candidata), encontrar uma transformação T sobre X , tal que $T(X)$ seja um subconjunto de S .

Em outras palavras, encontrar uma transformação que verifique que X é um subconjunto de S . Assim, pode-se concluir que os dados X e S pertencentes a Fragmento e Candidata respectivamente, são dados adquiridos de imagens distintas que possivelmente representam dados da mesma impressão digital.

4.2 Metodologia adotada para encontrar a solução

Se existir T então será assumido aqui que X é um subconjunto de S . Na tentativa de encontrar T (e informar que possivelmente os pontos adquiridos representam a mesma impressão digital), uma metodologia foi projetada e é descrita abaixo.

Podem existir várias soluções, ou seja, várias transformações T , que comprovem que X é um subconjunto de S . Para encontrar estas possíveis soluções, a partir dos conjuntos de pontos X e S (que são as minúcias pré-localizadas nas imagens das impressões digitais), dois grafos, $G_F=(V_F,E_F)$ e $G_C=(V_C,E_C)$ são construídos. V_F é o conjunto de pontos X e V_C é S . O conjunto de arestas E_F e E_C representam uma particular vizinhança entre o conjunto de pontos X e S . Estes grafos além de representarem as impressões digitais são fundamentais para o processo de comparação, uma vez que é através deles que são estabelecidas por meio das arestas as relações de vizinhança entre os pontos dos conjuntos.

A partir dos dois grafos construídos, G_F e G_C , pretende-se verificar se G_F é um subgrafo de G_C , comprovando em outras palavras que X é um subconjunto de S .

Ou seja,

$$\text{Se } G_F=(V_F,E_F) \subseteq G_C=(V_C,E_C) \rightarrow \exists T \mid T(X) \subseteq S$$

Perceba que se $G_F \subseteq G_C$, então existe uma bijeção entre $V_F = X$ e um subconjunto de $V_C = S$, a qual define um isomorfismo. Como o isomorfismo leva em conta a distância euclidiana, os grafos são considerados com pesos, então temos uma transformação que leva X à um subconjunto de S ($T(X) \subseteq S$).

Para verificar se G_F é um subgrafo de G_C , validando a metodologia proposta, o isomorfismo sobre os grafos é aplicado. No isomorfismo abordado para este trabalho, além da topologia dos grafos, a geometria dos pontos que constituem as suas listas de vértices também são analisados no processo de comparação.

Um esquema que define graficamente o problema e a metodologia adotada no presente trabalho pode ser visualizado na Figura 6:

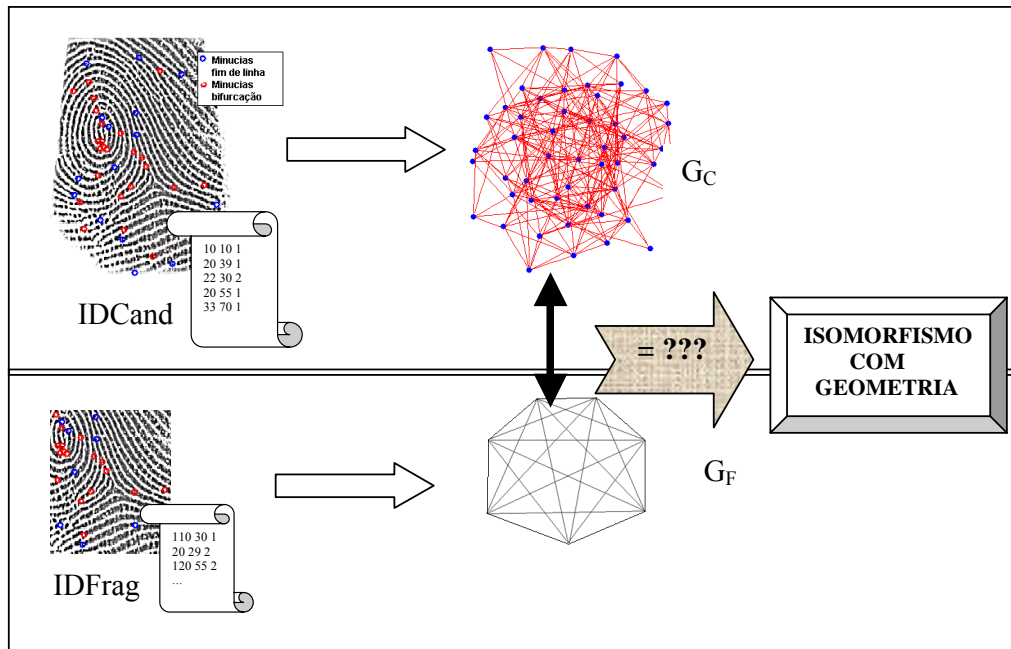


Figura 6 – Resumo da Metodologia proposta

4.3 Resolução do problema

Nesta seção são fornecidos maiores detalhes sobre a metodologia definida para resolver o problema de comparação de impressões digitais e o algoritmo implementado para calcular o isomorfismo de grafos com geometria. Inicialmente, as macro atividades com relação ao desenvolvimento deste trabalho são apresentadas e posteriormente o algoritmo de cálculo de isomorfismo propriamente dito juntamente com sua análise de desempenho são apresentados.

4.3.1 Encontrar minúcias nas imagens de impressão digital

Para a comparação das impressões digitais é necessário fornecer para o sistema alguma informação sobre cada ID como entrada. Grande parte dos trabalhos de comparação de impressões digitais analisados geralmente utilizam as minúcias do tipo bifurcação e final de linha encontradas como representação de cada impressão digital na comparação, uma vez que estas são as minúcias frequentemente e mais facilmente detectáveis nas imagens. Em um sistema de reconhecimento de impressões digitais estas características são previamente encontradas por um algoritmo de processamento de imagens. Como o foco deste trabalho não

é o processamento de imagens e sim o processo pós-processamento que aqui é a comparação baseada em grafos, a busca por estas características é feita de forma manual por um usuário, que analisa visualmente duas imagens de impressão digital a serem comparadas e escolhe as minúcias a serem utilizadas no processo de comparação (informando se a mesma é final de linha ou bifurcação).

4.3.2 Geração do grafo

Tendo os dados (minúcias) que constituem uma impressão digital, é construído um grafo que a representa. As minúcias de uma impressão digital formam o conjunto de vértices do grafo, V , e as arestas, E , (que dependem do tipo de grafo construído) representam a relação de vizinhança entre as minúcias. A seção 4.5 fornece os compromissos a serem assumidos quando se escolher os tipos dos grafos para que os mesmos contribuam no processo de comparação.

4.3.3 Cálculo do isomorfismo

Após a construção dos grafos G_F e G_C , a partir dos dados adquiridos das respectivas imagens Fragmento e Candidata, o processo de comparação é realizado por meio de sucessivas associações de vértices de G_F com vértices do grafo G_C . Estas associações são realizadas neste por meio do cálculo de isomorfismo de grafos. Inicialmente é definida uma transformação de uma aresta qualquer (denominada de aresta base) de G_F para uma aresta arbitrária de G_C onde uma pré-associação de seus vértices é realizada. A partir desta transformação criada, os encaixes vão sendo verificados de acordo com a análise de $OrdemF$ de G_F – definida percorrendo-se as arestas de G_F – que estabelecem juntamente com a pré-associação dos vértices de uma aresta arbitrária de G_C , realizada anteriormente, a ordem para se percorrer G_C .

A comparação de grafos retorna verdadeira quanto a árvore geradora de G_F é isomorfa à um subgrafo de G_C . Por isso ao definir G_F , caso os dados representem a mesma impressão digital, deve-se garantir que as arestas deste estejam presentes em G_C .

O algoritmo é exaustivo (exato), porém pode ter erros devidos aos arredondamentos numéricos da transformação geométrica, por isso no momento da comparação geométrica dos vértices um limiar de ponderação de diferenças geométricas é utilizado.

4.4 Portugol do algoritmo

Nesta seção o algoritmo desenvolvido para o cálculo de isomorfismo entre dois grafos é descrito. Este algoritmo busca encontrar um mapeamento das minúcias da imagem do fragmento, representadas aqui pelo conjunto de pontos X (base para construir o grafo G_F) e as minúcias da imagem candidata, conjunto de pontos S que formam o grafo G_C .

Sendo $G_F=(V_F,E_F)$ e $G_C=(V_C,E_C)$, construídos a partir das minúcias detectadas em cada imagem e suas respectivas listas de vértices, lista de arestas e matriz de adjacência, é necessário definir uma ordem de busca nos vértices de G_F . Esta ordem pode ser uma busca em largura ou profundidade no grafo, por exemplo. No presente trabalho, uma varredura em largura foi executada em G_F denominada aqui por *OrdemF*. Paralelamente à *OrdemF*, é mantida uma lista, denominada *PaiF*. Esta lista contém o pai, que é o predecessor de cada vértice na busca e formação de *OrdemF*.

A Figura 7 e Figura 8 apresentam o portugol do algoritmo de isomorfismo de G_F com G_C implementado neste trabalho. MAP é a matriz de mapeamento que informa para cada vértice V_F de G_F , qual é o vértice V_C que está mapeado em G_C .

```
Input: Grafo  $G_F$ , Grafo  $G_C$ 
Output: Array MAP
1. éIsomorfo
2. Escolha uma Aresta  $a=ij \in E_F$  ..... (aresta base)
3. Para cada Aresta  $b=kl \in E_C$ 
4. inicio
5.   MT = getMatriz_TransformaçãoT(a,b)
6.   Pré-associa: MAP[i]<-k e MAP[j]<-l
7.   Marca_Visitado(k,l);
8.   Testar_resto(MT,MAP, $G_F,G_C$ )
9.   Se terminou de testar e OK, então retorna MAP
10. fim.
```

Figura 7 – Portugol: núcleo da resolução do problema

```

Input: Array MT, Array MAP (parcial), Grafo  $G_F$ , Grafo  $G_C$ 
Output: Array MAP (atualizado) ou falha
11. Rotina: Testar_resto(Array MT, Array MAP, Grafo  $G_F$ , Grafo  $G_C$ )
12. Para i de 3 a  $|V_F|$ 
13. inicio
14.   w <- OrdemF(i);
15.   z <- MAP[PaiF(w)]
16.   ListaVizF <- getVizinhos( $G_F$ , z)
17.   Para cada vértice vizinho y não marcado do z (ListaVizF)
18.   inicio
19.     casouW <- Testa_MatrizTransformacao(T, w, y);
20.     Se casouW
21.     inicio
22.       MAP[w] <- y
23.       marca y como visitado
24.     fim
25.   fim
26. fim

```

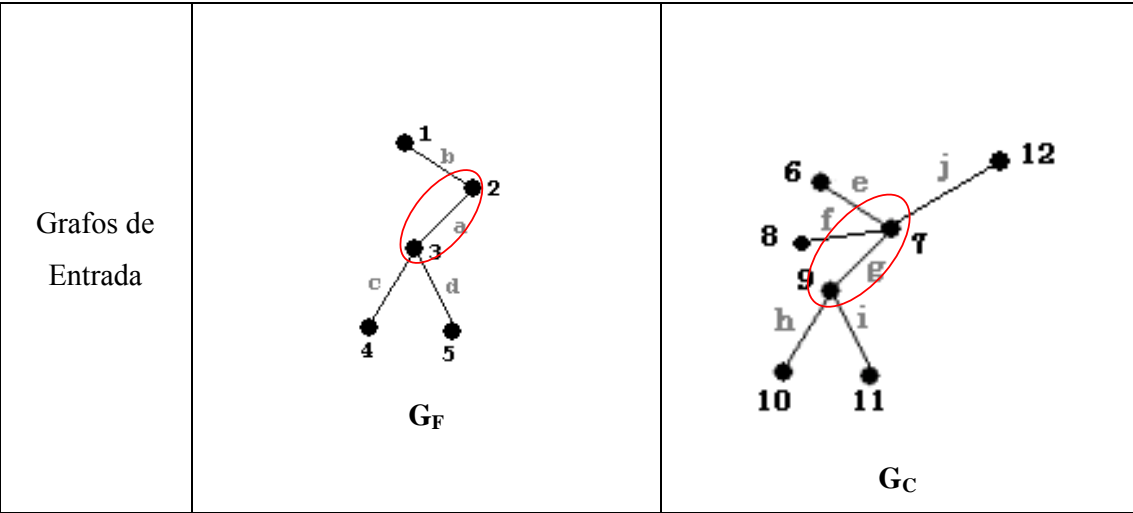
Figura 8 – Portugol: continuação do teste de isomorfismo

A rotina `getMatrizTransformação(Aresta1, Aresta2)`, retorna uma matriz de mapeamento da geometria dos vértices da aresta base E_F para os vértices de outra aresta arbitrária E_C . Este mapeamento consiste de uma transformação rígida (translação e rotação) com escala definido numa matriz 3x3 com coordenadas homogêneas. Esta matriz de mapeamento indica uma transformação T que será aplicada a cada novo vértice de V_F e comparado a um determinado vértice de V_C para verificar se suas geometrias se encaixam. Esta aplicação de T e posterior comparação de geometria utiliza uma tolerância para contornar os erros por arredondamento durante a definição de T e neste trabalho é denominado limiar geométrico.

Já a rotina `getVizinhos(Grafo G, Vértice V)` busca por meio da matriz adjacência de um grafo G , a lista de vértices adjacentes de V passado por parâmetro.

Resumidamente o isomorfismo é executado da seguinte forma: a partir da pré-associação da aresta base $a=ij$ de E_F de com $b=kl$ de E_C de G_C , a busca segue sobre os demais vértices V_F de acordo com a lista `OrdemF`. Assim, w (próximo elemento de `OrdemF`) é visitado. Procura-se o mapeamento z já feito para seu predecessor e busca-se pelos y vizinhos

de z . A partir daí, é procurado par para os vértices w nos vértices y e vizinhos sem par do z . Para cada novo par de vértices $\{w,y\}$, Testa_MatrizTransformação, aplica a matriz transformação T (calculada anteriormente através do cálculo da transformação geométrica de E_F de G_F em E_C de G_C) no ponto de w e verifica se y é par do w em G_C . Caso w seja o par de y , o loop continua para os próximos w vértices da listaOrdemF, sempre buscando para ser seu par na matriz MAP, o último vértice associado de G_C . Caso Testa_MatrizTransformação retorne falso, então o algoritmo retorna à linha 3, onde uma nova aresta arbitrária de G_C é escolhida e uma nova matriz transformação T é construída. A Figura 9 fornece um exemplo sobre os primeiros passos para o algoritmo de isomorfismo implementado.



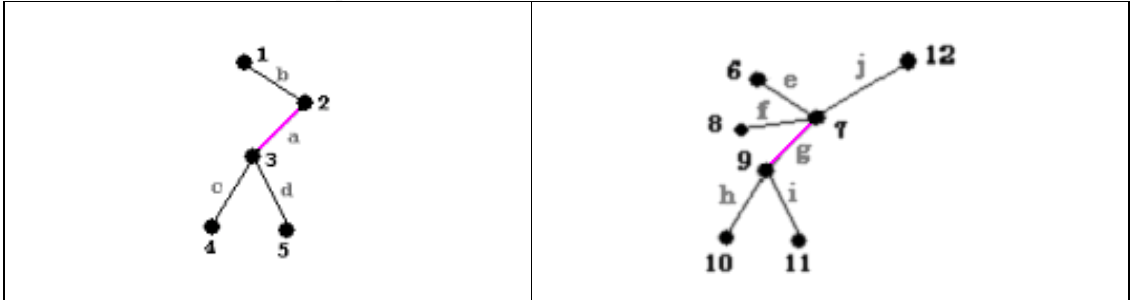
Cenário inicial:

VF = {1, 2, 3, 4, 5}
 EF = {a, b, c, d}
 OrdemF = {2, 3, 1, 4, 5}
 PaiF[1] = 2
 PaiF[2] = null
 PaiF[3] = 2
 PaiF[4] = 3
 PaiF[5] = 3

VC = {6, 7, 8, 9, 10, 11, 12}
 EF = {g, e, f, h, i, j}
 MAP = null

1º Passo: Pré-mapeamento (linhas 2 à 7 do algoritmo)

* Transformação: Aresta “a” para aresta “g”



* Mapeando aresta “a” com “g”

MAP[1] = null
MAP[2]=7
MAP[3]=9
MAP[4] = null
MAP[5] = null
Vizitados = {7, 9}

2º Passo: rotina Testar_Resto (linhas 14 à 16 do algoritmo)

* Pegando próximo vértice na lista ordem

i = 3;
w = 1;
PaiF(w)=2
z = 7
ListaVizF = {8, 6, 12}

3º Passo: linhas 17 à 25 do algoritmo

y = 8;
casouW = false
y = 6;
casouW = true

MAP[1] = 6
MAP[2]=7
MAP[3]=9
MAP[4] = null
MAP[5] = null
Vizitados = {7, 9, 6}

4º Passo: linhas 14 à 16 do algoritmo

* Pegando próximo vértice na lista ordem

i = 4;
w = 4;
PaiF(w)=3
z = 9
ListaVizF = {10, 11}

5º Passo: linhas 17 à 25 do algoritmo

y = 10;
casouW = true

MAP[1] = 6
MAP[2]=7
MAP[3]=9
MAP[4] = 10
MAP[5] = null
Vizitados = {7, 9, 6, 10}

6º Passo: linhas 14 à 16 do algoritmo

* Pegando próximo vértice na lista ordem

i = 5;
w = 5;
PaiF(w)=3
z = 9
ListaVizF = {10, 11}

7º Passo: linhas 17 à 25 do algoritmo

y = 11;
casouW = true

MAP[1] = 6
MAP[2]=7
MAP[3]=9
MAP[4] = 10
MAP[5] = 11
Vizitados = {7, 9, 6, 10, 11}

Figura 9 – Exemplo de execução do algoritmo

Perceba que para este algoritmo, o isomorfismo é realizado através da procura de um subgrafo de G_C que seja isomorfo à árvore geradora de G_F . Embora o resultado não seja o isomorfismo dos grafos, este continua mantendo a propriedade desejada que é $T(X) \subseteq S$.

Durante a execução da rotina *éIsomorfo*, uma aresta de G_F , é escolhida para ser transformada e comparada com cada aresta de G_C . Enquanto não esgotarem-se as comparações dos vértices de G_F , novas arestas são escolhidas em G_C , na tentativa de novas transformações geométricas, ou seja, quanto mais vezes o loop da linha 3 tiver que ser executado, maior será o tempo para executar o algoritmo de *Testar_resto*, que depende diretamente do número de vértices de G_F e do número de arestas de G_C .

A seção 4.6 detalha a complexidade deste algoritmo que é $|V_F| * |V_C| * |E_C|$.

4.5 Características necessárias para os grafos G_C e G_F

A idéia principal de um algoritmo de cálculo do isomorfismo de grafos aplicado neste trabalho é buscar um mapeamento f de cada vértice V_F de um grafo G_F para os vértices V_C de um grafo G_C , retornando como resultado se um isomorfismo foi encontrado ou não.

Um algoritmo de isomorfismo em geral, procura o encaixe de acordo com a topologia do grafo, analisando a vizinhança para cada vértice de um grafo G_F juntamente com a vizinhança de G .

Em se tratando de analisar também a geometria - de fundamental importância para a comparação de impressões digitais - é necessário além de analisar a topologia no cálculo do isomorfismo, encontrar uma transformação T que permita o encaixe geométrico dos grafos a serem analisados. Encontrar a transformação geométrica é importante, pois as imagens nem sempre são bem comportadas, podendo cada aquisição ser de diferente escala, ou uma imagem é transladada ou rotacionada em relação à outra. Por este motivo, para o presente trabalho, a matriz T a ser encontrada não é qualquer, podendo envolver translação, rotação e escala dos pontos. Vale também lembrar que para o caso de comparação de impressões digitais dois vértices quaisquer só poderão ser mapeados um para o outro se representarem a mesma minúcia.

A definição dos tipos de grafos criados para representar cada imagem é de fundamental importância. Não há uma regra que defina para cada problema computacional qual o grafo mais adequado para se encontrar a solução. Para este problema de comparação de impressões digitais algumas considerações prévias sobre os tipos de grafos são de grande

relevância. Na sequência são listados os compromissos a serem assumidos no momento de se escolher os tipos de grafos para representarem a impressão digital Fragmento e Candidata:

- i) O grafo completo (que contém o número máximo de arestas) é um grafo ideal para representar as impressões digitais, pois ele mostra todos os possíveis relacionamentos entre os vértices que o formam. Por este motivo se G_F e G_C são grafos completos então uma solução positiva é sempre encontrada caso as imagens analisadas representem a mesma impressão digital, sendo esta a grande vantagem de se utilizar este tipo de grafo. Por outro lado, o número de testes para serem realizados ao se testar um grafo deste tipo no isomorfismo é extremamente grande e custoso;
- ii) Diminuir o número de arestas de G_C é uma idéia boa, uma vez que o espaço de busca é menor, porém se o número de arestas de G_F for aumentado ao mesmo tempo pode trazer desvantagens. Por exemplo, se G_C é um grafo estrela e G_F um grafo completo, podemos intuitivamente perceber que muitas arestas de G_F faltarão no momento do cálculo do isomorfismo, e, mesmo se as imagens representem a mesma impressão digital provavelmente, o isomorfismo do Fragmento com a Candidata não será encontrado, pois é necessário encontrar todos os vértices de G_F e alguns de seus relacionamentos em G_C , sendo o isomorfismo para este trabalho considerado um sub-isomorfismo, ou seja, um isomorfismo de subgrafo;
- iii) Para melhorar o desempenho no momento da comparação, deve-se priorizar a otimização dos tipos de grafos gerados para representar as impressões digitais. Um grafo com poucos vértices e/ou arestas seria o ideal. O problema ao diminuir vértices para a comparação de impressões digitais implica em termos menos dados a serem utilizados no processo de comparação. Se o número de vértices for muito pequeno (principalmente em G_F), conseqüentemente o número de arestas também é menor e o número de falsos positivos pode aumentar (pode-se encontrar isomorfismo com outros grafos que não necessariamente representem a mesma impressão digital). Se o grafo G_C for um grafo completo tem-se neste grafo todas as possíveis soluções a serem testadas, sendo um compromisso garantir que G_F seja um grafo que tenha no mínimo os vértices e arestas que estão em G_C para o caso das imagens representarem a mesma impressão digital. Uma vez que esta condição esteja garantida, deve-se tentar diminuir o tamanho de G_C para, conseqüentemente, diminuir o tempo utilizado para a busca da solução. Para realizar melhorias, neste trabalho é proposto um isomorfismo com controle no tamanho das arestas de G_C que tem por objetivo realizar uma triagem das arestas a serem utilizadas na comparação. Esta otimização é discutida na seção 4.7.

O presente trabalho se propõe a analisar os resultados encontrados dependendo do tipo de grafo escolhido para realizar a comparação de impressões digitais, tanto para representar a imagem do Fragmento, como a imagem Candidata. Não é trivial encontrar os grafos $e(G_F)$ e $e(G_C)$ adequados à encontrar a solução, no decorrer do texto e dos experimentos pretende-se esclarecer estas dificuldades.

4.6 Análise do Algoritmo

O tempo necessário para determinar o isomorfismo entre os dois grafos depende diretamente do número de $|V_F|$, $|V_C|$ e $|E_C|$. Pode se dizer sucintamente que a complexidade do algoritmo de isomorfismo é $|V_F| * |V_C| * |E_C|$.

Analisando diretamente o algoritmo podemos levantar os seguintes detalhes:

i) A linha 2 é responsável por escolher uma aresta “base” do grafo da impressão fragmento, G_F . Esta aresta será analisada juntamente com cada aresta do grafo da impressão candidata G_C , por meio do loop que está escrito na linha 3. Quanto mais arestas tivermos em G_C enquanto um isomorfismo não é encontrado, maior o número de testes (cálculos de uma transformação T) a serem feitos.

ii) Enquanto não terminar de comparar todos os vértices de G_F e um isomorfismo não tiver sido encontrado, para a aresta base escolhida de G_F e cada aresta de G_C , a rotina `getMatrizTransformacao` calcula a transformação T necessária para transformar geometricamente as extremidades da aresta E_F de G_F nas extremidades da aresta E_C de G_C . Lembrando, que os grafos utilizados aqui não são dirigidos, portanto há 2 (duas) T possíveis de serem calculadas para cada par de arestas. Novamente, uma T só poderá ser calculada caso os vértices a serem pré-mapeados representem as mesmas minúcias. O tempo para calcular esta transformação T é constante, pois ela é executada uma vez para cada par de arestas E_F , E_C .

iii) A rotina `Testar_resto`, é a grande consumidora de tempo, pois para cada pré-associação da aresta de G_F , com aresta de G_C (linha 6), é gerada a Matriz de Transformação T e o algoritmo segue buscando os próximos vértices de G_F percorrendo a `listaOrdemF` (loop da linha 12). Quanto mais minúcias forem detectadas na impressão digital do fragmento, maior será o tamanho da lista `OrdemF`, que é a lista dos vértices do Fragmento (criada conforme suas arestas), o que implica na complexidade do loop desta linha.

iv) Para cada vértice w da lista OrdemF , seu pai é procurado, e verifica-se com qual vértice z do grafo G_C ele está casado. Encontrando-se z , busca-se quais são os vértices y de G_C adjacentes à z pelas linhas 16 e 17 (Figura 8) respectivamente. Vale lembrar que quanto mais arestas tiver G_C , mais custoso será o loop da linha 17, uma vez que ele depende diretamente do grau de G_C . Cada vez que é encontrado um par para o w (testando a matriz T para w e verificando se é igual à y , ou dentro de um limiar – tolerância - de erro), na busca para os vizinhos de y , o loop continua para o próximo vértice de G_F presente na lista OrdemF , caso não seja encontrado nenhum par, o algoritmo retorna para a linha 3, procurando uma próxima aresta de G_C , para refazer a transformação e prosseguir novamente nos testes.

4.7 Otimização proposta e implementada: isomorfismo com controle no tamanho das arestas

Como exposto anteriormente se o número de vértices de G_F for minimizado, pode-se encontrar muitos falsos positivos. Por este motivo, para se melhorar o desempenho no cálculo do isomorfismo, realizou-se um estudo para tentar minimizar o número de arestas de G_C a serem avaliadas no momento de se criar uma transformação T .

Analisando as imagens de impressões digitais de Fragmento e Candidata, observa-se que geralmente a imagem do fragmento é uma imagem com uma área menor onde um menor número de pontos é detectado e, conseqüentemente, o grafo que a representa possui menor ordem e grau que o grafo da imagem candidata.

Aproveitando-se deste detalhe, pode-se determinar um rótulo para cada aresta, de acordo com sua medida que pode ser dada pelas coordenadas geométricas de seus vértices, rotulando-se assim para este trabalho, cada aresta com o tamanho encontrado pela distância euclidiana dada pelas coordenadas de suas extremidades.

De acordo com esta pré-análise do tamanho da área útil das imagens relacionado ao tamanho das arestas no grafo que a representa, pode-se dizer que a menor aresta de G_F , m_F , freqüentemente tem um tamanho maior que a menor aresta de G_C , m_C , e que maior aresta de G_F , M_F , é menor que a maior aresta de G_C , M_C . Assim:

$ m_C \leq m_F \text{ e } M_F \leq M_C $
--

Com base nesta característica, uma proposta para diminuir o tempo de busca é limitar o espaço de busca da linha 3 do código (que busca par para a aresta de G_F nas arestas de G_C). O que propõe-se aqui é verificar a proporcionalidade P entre $|M_C|$ e $|M_F|$, desta forma $P = |M_C| / |M_F|$. Tendo esta constante P , podemos delimitar o limite superior de análise nas arestas de G_C , sendo que a aresta de maior tamanho de G_C a ser analisada na linha 3, deve possuir tamanho menor ou igual à:

$$P * |m_F|$$

Como T é uma transformação rígida construída a partir de uma aresta de G_F e uma de G_C , de modo a transformar a primeira na segunda, T é composta de uma escala que modifica o tamanho desta aresta. Ao construir T usando a maior aresta de G_F , M_F , com a maior aresta de G_C , M_C , a escala é $P = |M_C| / |M_F|$. Desta forma, a menor aresta de G_F , m_F , ficaria de tamanho $P * |m_F|$. Logo nenhuma aresta de G_C com tamanho maior que $P * |m_F|$ pode ser associada a m_F . A teoria desta formulação explicada aqui pode ser visualizada no quadro abaixo.

- 1) Dado que: $P = \frac{|M_C|}{|M_F|}$ (proporção máxima)
 - 2) Se $T(m_F) = a_C$ onde a_C é uma aresta qualquer de G_C ,
 - 3) Tem-se que a escala desta transformação é $= \frac{|a_C|}{|m_F|}$
 - 4) Logo, $\frac{|a_C|}{|m_F|} * |M_F| = |T(M_F)|$
- Será que $|T(M_F)| > |M_C|$?
- Se $|a_C| > P * |m_F|$
- De 4), temos que $|T(M_F)| = \frac{|a_C|}{|m_F|} * |M_F|$
- Assim, $|T(M_F)| > \frac{P * |m_F|}{|m_F|} * |M_F| = \frac{|M_C|}{|M_F|} * |M_F|$
- Logo, se $|a_C| > P * |m_F|$ então $|T(M_F)| > |M_C|$

Se T for construída usando m_F e uma aresta a_C , tal que $|a_C| > P * |m_F|$, então

$|T(M_F)| > |M_C|$ e a aresta M_F ficaria sem correspondência em G_C .

Na seção de experimentos será exposto um quadro demonstrando os resultados obtidos com a implementação desta otimização no cômputo do isomorfismo utilizando o controle no tamanho das arestas.

Para um sistema de reconhecimento de impressões digitais, basta que 1 (uma) relação que indique um isomorfismo encontrado entre o grafo da impressão fragmento e o grafo que representa a impressão candidata, para indicar que as imagens são de uma mesma impressão digital.

4.8 Estado da arte - trabalhos semelhantes

Muito já foi pesquisado sobre a comparação de grafos, podemos destacar aqui alguns dos trabalhos analisados:

- Ângelo Duarte, Claudia Tedesco e Daniel Lucena, [DUA2004] utilizaram grafos com a intenção de se construir uma estrutura eficiente de armazenamento das minúcias encontradas na impressão digital que contém dados referentes à distribuição, tipo da minúcia e relacionamento entre as minúcias. Foi criada uma lista de adjacência de vértices, onde cada lista $A(v)$ é denominada lista de adjacências do vértice v e contém os vértices w adjacentes a v em G . Os vértices, por sua vez, representam as minúcias encontradas na impressão digital e as arestas rotuladas contém a distância de V ao centro da impressão digital. O grafo está no formato de uma árvore. A comparação dos grafos que representam as impressões digitais é feita de maneira simples onde é comparado a posição (x,y) da minúcia armazenada no grafo, a distância do centro do grafo gerado e tipo de minúcia. Então é considerado um percentual de minúcias equivalentes para dois grafos. Infelizmente não foi encontrado esclarecimento de como o tratamento para imagens rotacionadas, transladadas e escaladas foi realizado;

- Álvaro Matos, [MAT1993] representou os padrões de impressão digital na forma de grafos, onde as minúcias e suas informações inerentes (tipo e quadrante) foram representadas através dos vértices e as relações entre as minúcias (contagem de linhas e orientação relativa) são representadas pelas arestas. Assim, o problema de comparação entre duas impressões digitais foi abstraído para o problema de isomorfismo entre grafos, com algumas diferenças importantes como: os vértices possuem tipos diferentes e assim, apenas vértices do mesmo tipo são mapeados; os vértices possuem quadrantes determinados pelos sistemas de referência descritos, e portanto, mesmo vértices do mesmo tipo só são mapeados se estiverem nos quadrantes respectivos; as arestas são valoradas e só podem ser mapeadas se

apresentarem seus vértices terminais já mapeados, contagem de linhas e orientações iguais (ou dentro de uma faixa pré-estabelecida); a exigência estrutural do isomorfismo, de que o número de arestas entre um par de vértices de um grafo G é o mesmo entre seus vértices correspondentes em H não é rígida. O que se busca é o maior número de arestas coincidentes. Dessa forma a comparação entre os grafos é estrutural, posicional e valorada, e a solução proposta foi a utilização de um algoritmo eficiente que realize esta comparação respeitando as características do problema já descritas;

- D. K Isenor, [ISE1986] utilizou um algoritmo insensível à imperfeições que consistia da criação de um grafo para a comparação de impressões digitais no qual os nós representam as linhas da impressão digital e arestas representam as linhas que são vizinhas ou linhas que se dividem. Desta forma, a estrutura do grafo representa as relações topológicas dentro da impressão digital. Um algoritmo de três passos (particionamento, refinamento e pontuação) foi empregado para encontrar a comparação entre um par de impressões digitais.

5. DESENVOLVIMENTO

Este capítulo descreve os detalhes de implementação deste trabalho.

5.1 Linguagem

Principalmente pelo fato de ser independente de plataforma, Java foi a linguagem escolhida para o desenvolvimento deste trabalho. Desta forma, é possível desenvolver uma aplicação sem se preocupar com o sistema operacional no qual ele será executado, uma vez que existe uma máquina virtual Java para cada sistema operacional que faz a ponte entre o código Java compilado (bytecode) e o código nativo da máquina. Outro item que incentivou o desenvolvimento em Java é a quantidade de APIs que existem disponíveis para agilizar a implementação. A implementação é orientada a objetos seguindo o paradigma de Java.

5.2 Implementação

Para auxiliar nos testes um ambiente foi desenvolvido e será apresentado na próxima seção. Como já exposto anteriormente, a implementação seguiu as normas da Orientação à Objetos e por isso, as seguintes classes foram desenvolvidas:

- Ponto: representa as coordenadas geométricas (x,y) de cada minúcia encontrada na impressão digital;
- Minúcia: representa uma característica encontrada na impressão digital. É formada por um ponto (localização geométrica) e o tipo da minúcia encontrada na impressão digital, que pode ser um final de linha (FL) ou uma bifurcação (B);
- Vértice: representa o vértice do grafo e sua informação é a minúcia encontrada na impressão digital;
- Aresta: formada por dois vértices representa uma relação de adjacência. A aresta pode ter um peso associado (vide objeto `PesoAresta`) que é utilizado para fins de consulta somente nos testes quando foi calculado o isomorfismo com controle no tamanho das arestas;

- Grafo: representa o grafo, com sua lista de Vértices e Arestas e sua relação de vizinhança através da matriz de adjacência. Esta classe possui o método com o algoritmo principal para o cálculo do isomorfismo;
- Impressão Digital: representa uma impressão digital. A imagem a que está vinculada, as minúcias encontradas e o seu grafo associado que tem como finalidade representá-la no momento da comparação com outra impressão digital;
- PesoAresta: representa o peso de uma aresta, que para este trabalho é a distância euclidiana entre seus dois vértices. Esta medida tem por finalidade indicar o tamanho de uma aresta no qual este está associado;
- MatrizTransformação: classe auxiliar que representa a matriz de transformação rígida com escala entre uma aresta de um grafo G_F e os vértices de uma aresta de um grafo G_C ;
- Controle: é a classe principal responsável pelo gerenciamento das chamadas aos métodos corretos dentro das opções solicitadas pelo usuário no programa implementado;
- Tela: classe auxiliar na qual estão implementados os controles da interface gráfica;
- LogApp: responsável por manter um controle das comparações realizadas no algoritmo, para fins de visualização do passo a passo durante a execução, resultados e estimativas obtidas;
- Estatística: para fins de manter um controle sobre as estimativas, esta classe mantém informações sobre número de vértices e arestas dos grafos, número de comparações realizadas e o tempo para se encontrar uma solução para o problema;
- CasamentoG: mantém o mapeamento resultante do cálculo do isomorfismo de dois grafos. Para cada casamento encontrado, sua estimativa está associada.

O macro diagrama de classes do sistema pode ser visualizado na Figura 10.

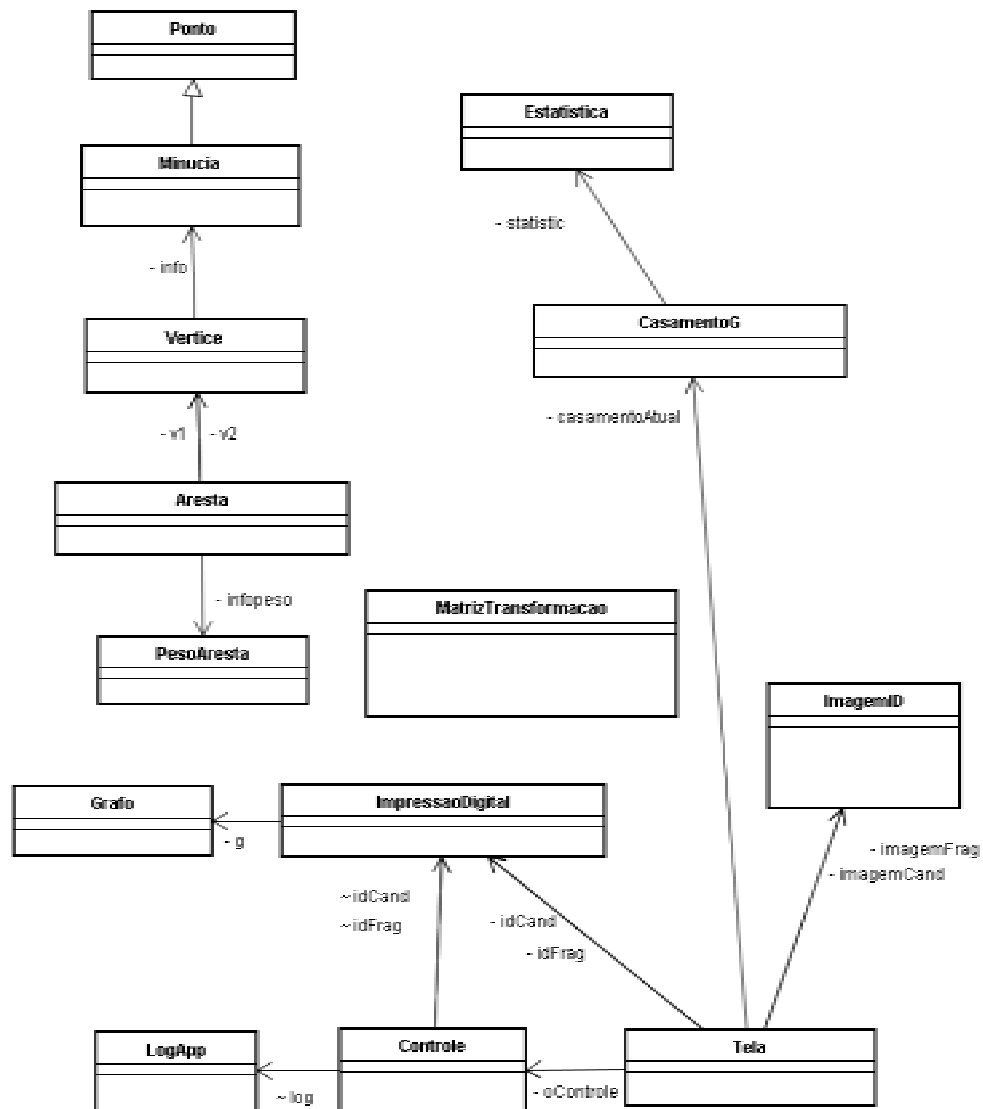


Figura 10 – Diagrama de classes da implementação realizada

5.3 A ferramenta de testes

A Figura 11 apresenta um *snapshot* da ferramenta desenvolvida para que os experimentos fossem realizados e visualizados.

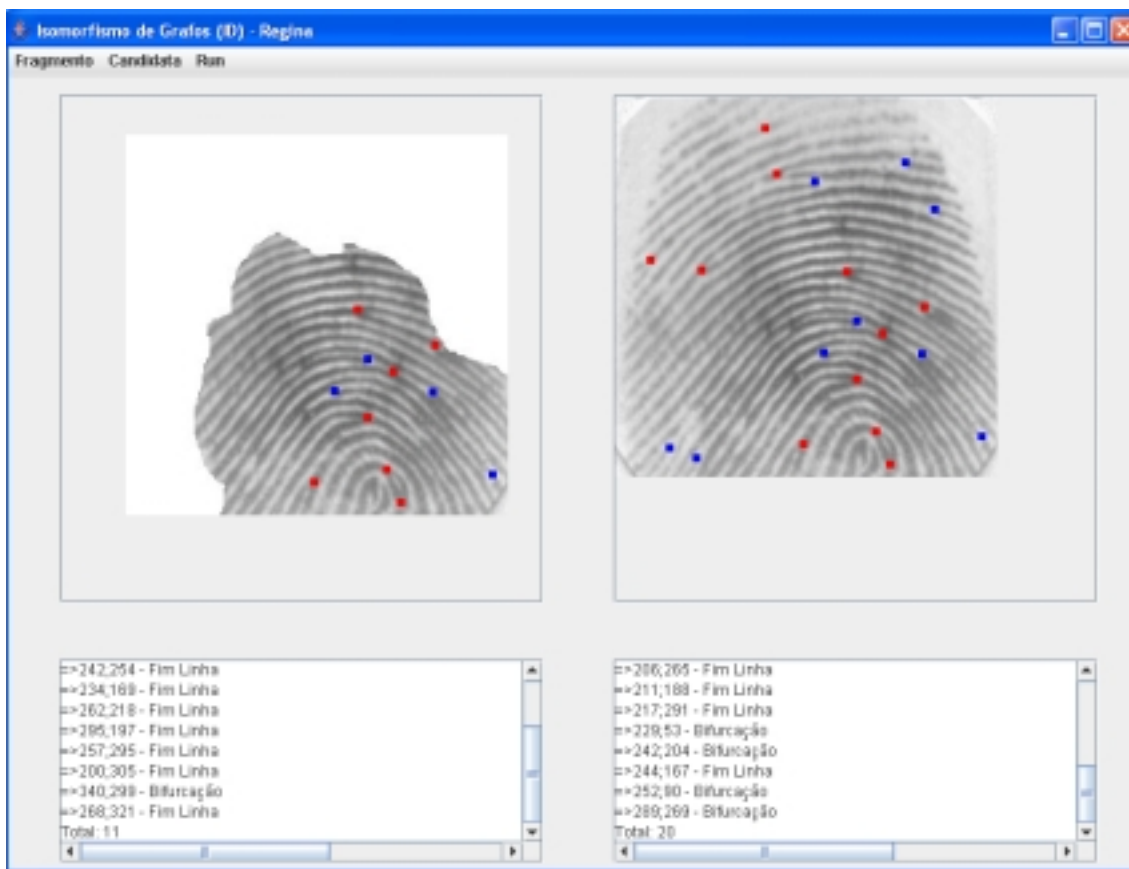


Figura 11 – *Snapshot* da ferramenta utilizada para realizar os experimentos

Esta ferramenta permite abrir as imagens de impressão digital desejadas para realizar a comparação a partir dos menus *Fragmento* e *Candidata*. No momento em que as imagens são carregadas, suas minúcias pré-detectadas que estão em arquivos anexos (conforme Tabela 1) às imagens são também lidas e estas características são mostradas em destaque nas imagens. As minúcias do tipo final de linha estão em destaque na cor azul e as do tipo bifurcação na cor vermelha.

CoordenadaX ₁	CoordenadaY ₁	TipoMinúcia ₁
CoordenadaX ₂	CoordenadaY ₂	TipoMinúcia ₂
...
CoordenadaX _n	CoordenadaY _n	TipoMinúcia _n

Tabela 1 - Layout do arquivo com minúcias pré-detectadas

Se o usuário desejar desconsiderar alguma minúcia da impressão do fragmento no momento da comparação, basta dar um clique sobre a mesma. Se quiser adicioná-la, basta

clicar no local da imagem desejado com o botão direito do mouse, para incluir uma minúcia do tipo final de linha e caso deseje incluir uma bifurcação, deve-se manter pressionado a tecla “shift” no momento do clique.

Através do menu Fragmento é possível realizar transformações geométricas nas imagens através dos submenus: Translação, Rotação e Escala.

O próximo passo para a realização do experimento é definir os grafos para cada impressão digital. As minúcias de cada impressão digital serão os vértices de cada grafo. A relação de vizinhança variará de acordo com o tipo de grafo escolhido. A escolha do tipo de grafo deve ser feita acessando os menus Fragmento e Candidata.

Tanto para a imagem do tipo fragmento (imagem à esquerda) como para a imagem da impressão candidata (imagem à direita) foram disponibilizados os grafos completo, estrela (com centro randômico) e bipartite completo.

O grafo estrela é um grafo bipartido de n vértices que possui um conjunto independente com 1 (um) único vértice e outro com $n-1$ vértices. O vértice isolado é denominado vértice central e nos experimentos realizados para este trabalho tem a intenção de simular o centro da impressão digital. O grafo bipartite é um grafo no qual o conjunto de vértices V pode ser particionado entre dois conjuntos não vazios V_1 e V_2 e cada aresta de G indica uma relação entre um vértice de V_1 e um vértice de V_2 (V_1 pode representar o conjunto de vértices que representam minúcias do tipo final de linha e V_2 do tipo bifurcação, por exemplo). O grafo completo é o grafo onde cada par de vértices distintos são adjacentes.

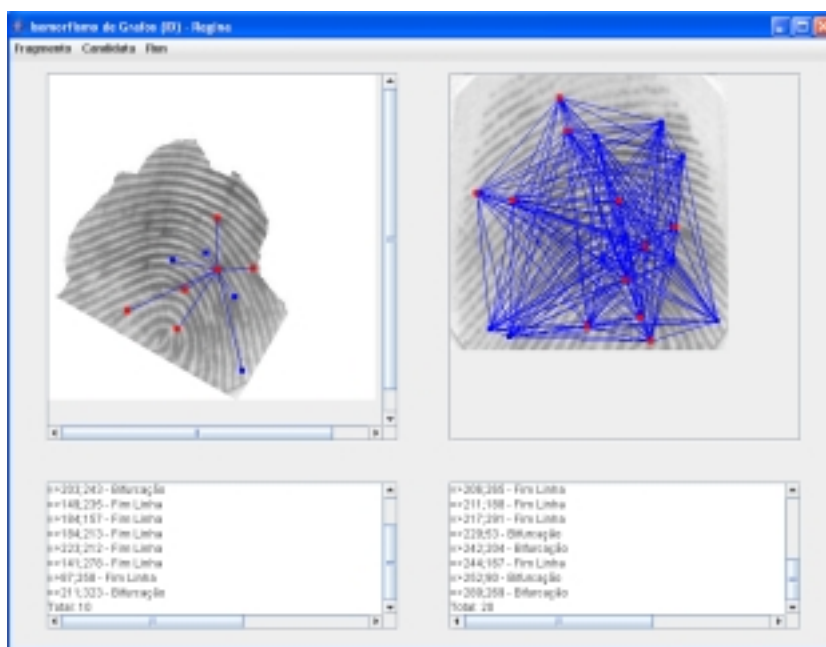


Figura 12 – Grafos criados para as impressões digitais

A Figura 12 mostra as impressões digitais com seus respectivos grafos construídos na interface desenvolvida. Para realizar a comparação, principal objetivo deste trabalho, basta acessar o menu “Run” e escolher a opção “Comparar” para realizar o cálculo do isomorfismo normalmente sem analisar os pesos (tamanho) das arestas ou “Comparar menos Arestas” para adicionar ao cálculo do isomorfismo a condição de se analisar o tamanho das arestas, visando realizar um menor número de comparações até encontrar uma relação de isomorfismo entre os grafos, caso a mesma exista.

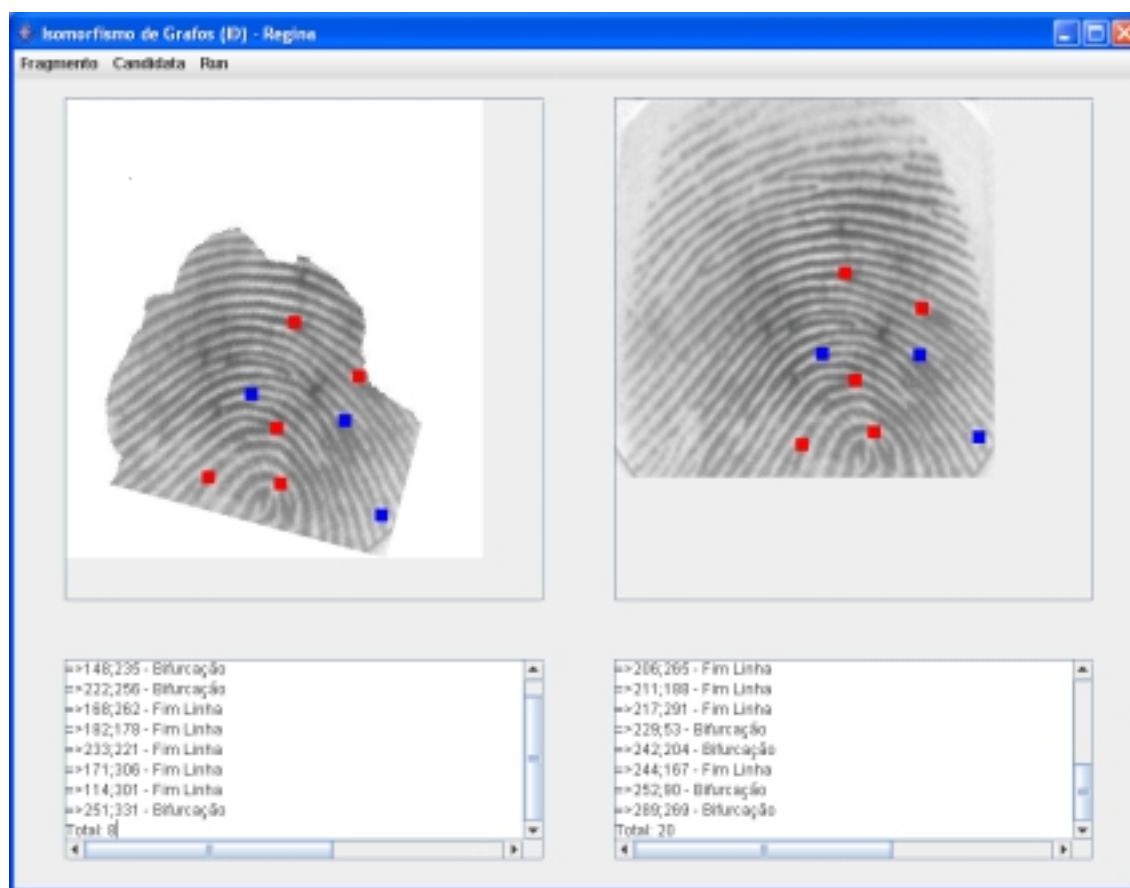


Figura 13 – Resultado da comparação com vértices que foram ‘casados’ em destaque

Após a comparação ser realizada os vértices que casaram aparecem em destaque (Figura 13) um pequeno resumo da comparação é mostrado, conforme a tabela abaixo (Tabela 2):

~o*o*o*o*o*o*o*o*o~
Achado 1 casamentos!
~o*o*o*o*o*o*o*o*o~

Mapeamento:

VFrag: 6 VCand: 14
VFrag: 1 VCand: 11
VFrag: 2 VCand: 9
VFrag: 3 VCand: 17
VFrag: 4 VCand: 12
VFrag: 5 VCand: 10
VFrag: 7 VCand: 18
VFrag: 8 VCand: 13
VFrag: 9 VCand: 7
VFrag: 10 VCand: 20
Estatística:

Nº Vértices do Frag: 10
Nº Arestas do Frag: 9
Nº Vértices da Cand: 20
Nº Arestas da Cand: 190
Nº Arestas Cand - Analisadas: 2
Nº Arestas Cand - Rejeitadas: 0
Nº de Comparações Geométricas Realizadas: 99
Tempo de execução: 50ms

Tabela 2 - Resumo da Comparação

Esta tabela nos mostra que para os grafos analisados, foi encontrado 1 casamento, ou seja 1 relação de isomorfismo. Em seguida, o mapeamento dos vértices da candidata para o fragmento são listados.

A estatística desta análise também é mostrada, informando o número de vértices e arestas do grafo da impressão digital Fragmento e da impressão digital Candidata, o número de arestas Candidatas analisadas até encontrar o mapeamento, e, caso haja alguma rejeição por motivo de tamanho da aresta, este também será informado no campo Arestas Rejeitadas. O número de comparações geométricas realizadas e o tempo de execução (em milissegundos) também são mostrados.

Para uma análise mais completa de como suas execuções procederam, é disponibilizado um arquivo de log, cuja pequena parte dele pode ser visualizado na Figura 14. Este arquivo de log, tem por finalidade mostrar com detalhes do que ocorreu durante a execução do cálculo do isomorfismo, tais como: quando aconteceu um casamento de vértices, quando foi necessário gerar uma nova transformação e quais os vértices e arestas da candidata que foram confrontados com vértices e arestas, respectivamente do fragmento.

31	AWT-EventQueue-0	DEBUG	LogApp	(Aresta : F: 1) - Vértices (u,v) de F: 0 e 1
31	AWT-EventQueue-0	DEBUG	LogApp	(Aresta : G: 148)
31	AWT-EventQueue-0	DEBUG	LogApp	(Aresta : C: 148) - Vértices (j,k) de C: 11 e 14
41	AWT-EventQueue-0	DEBUG	LogApp	Pol-MAP: 0 <-> 11
41	AWT-EventQueue-0	DEBUG	LogApp	Pol-MAP: 1 <-> 14
41	AWT-EventQueue-0	DEBUG	LogApp	Escolhendo outro de FI
41	AWT-EventQueue-0	DEBUG	LogApp	Escolhendo outro de FI
41	AWT-EventQueue-0	DEBUG	LogApp	Novo vértice (u) - frag a ser associado: 2
41	AWT-EventQueue-0	DEBUG	LogApp	1º pretendente. (y): 1
41	AWT-EventQueue-0	DEBUG	LogApp	2º pretendente. (y): 2
41	AWT-EventQueue-0	DEBUG	LogApp	3º pretendente. (y): 3
41	AWT-EventQueue-0	DEBUG	LogApp	4º pretendente. (y): 4
41	AWT-EventQueue-0	DEBUG	LogApp	5º pretendente. (y): 5
41	AWT-EventQueue-0	DEBUG	LogApp	6º pretendente. (y): 6
41	AWT-EventQueue-0	DEBUG	LogApp	7º pretendente. (y): 7
41	AWT-EventQueue-0	DEBUG	LogApp	8º pretendente. (y): 8
41	AWT-EventQueue-0	DEBUG	LogApp	9º pretendente. (y): 9
41	AWT-EventQueue-0	DEBUG	LogApp	10º pretendente. (y): 10
51	AWT-EventQueue-0	DEBUG	LogApp	11º pretendente. (y): 12
51	AWT-EventQueue-0	DEBUG	LogApp	12º pretendente. (y): 13
51	AWT-EventQueue-0	DEBUG	LogApp	13º pretendente. (y): 15
51	AWT-EventQueue-0	DEBUG	LogApp	14º pretendente. (y): 16
51	AWT-EventQueue-0	DEBUG	LogApp	15º pretendente. (y): 17
51	AWT-EventQueue-0	DEBUG	LogApp	16º pretendente. (y): 18

Figura 14 – Arquivo de log mantido durante o cálculo do isomorfismo

6. EXPERIMENTOS REALIZADOS E RESULTADOS ALCANÇADOS

Este capítulo tem como objetivo apresentar alguns experimentos realizados para comprovar a eficácia da metodologia proposta e desenvolvida neste trabalho, comentar sobre os mesmos e explanar sobre os resultados alcançados. Os experimentos estão divididos de acordo com os tipos de grafos escolhidos para a dupla Fragmento e Candidata: Estrela, Bipartite e Completo. Para cada experimento, e, conseqüentemente, tipos de grafos escolhidos, são apresentados os testes realizados computando-se o isomorfismo sem e com o controle no tamanho das arestas. O isomorfismo com controle no tamanho das arestas é uma proposta deste trabalho explanada e discutida na seção 4.7. Este é um isomorfismo que realiza uma análise prévia no tamanho das arestas de G_C , antes da construção da transformação T , que tem por objetivo encontrar a solução (relação de isomorfismo), caso exista, de forma mais ágil, se comparado ao sem o controle no tamanho das arestas.

6.1 Experimentos

Nesta seção são apresentados 5 (cinco) experimentos. Para os primeiros 4 (quatro) experimentos apresentados aqui, as mesmas imagens e minúcias detectadas foram utilizadas como dados de entrada. Estes dados de entrada, que são as posições geométricas (x,y) de cada minúcia, seu tipo e o vértice que a representa, podem ser visualizados na Tabela 3.

Impressão Digital	Fragmento: 13 minúcias	Candidata: 32 minúcias	
=>Posição Geométrica (x,y) Tipo Minúcia Rótulo do Vértice	=>231,195 - FL (V ₁) =>179,176 - FL (V ₂) =>211,245 - FL (V ₃) =>210,303 - FL (V ₄) =>91,281 - B (V ₅) =>216,160 - B (V ₆) =>208,222 - B (V ₇) =>178,203 - B (V ₈) =>129,195 - B (V ₉) =>182,309 - B (V ₁₀) =>279,157 - FL (V ₁₁) =>204,130 - FL (V ₁₂) =>127,291 - B (V ₁₃)	=>210,122 - FL (V ₁) =>159,59 - FL (V ₂) =>189,62 - FL (V ₃) =>155,118 - FL (V ₄) =>203,176 - FL (V ₅) =>217,232 - FL (V ₆) =>161,264 - FL (V ₇) =>57,78 - B (V ₈) =>192,52 - B (V ₉) =>258,151 - B (V ₁₀) =>250,121 - B (V ₁₁) =>97,242 - B (V ₁₂) =>186,93 - B (V ₁₃) =>195,154 - B (V ₁₄) =>161,144 - B (V ₁₅) =>246,73 - FL (V ₁₆)	=>270,88 - FL (V ₁₇) =>246,235 - B (V ₁₈) =>65,190 - FL (V ₁₉) =>85,200 - FL (V ₂₀) =>111,149 - B (V ₂₁) =>192,245 - B (V ₂₂) =>207,271 - B (V ₂₃) =>261,223 - B (V ₂₄) =>239,35 - B (V ₂₅) =>195,24 - FL (V ₂₆) =>62,34 - B (V ₂₇) =>40,114 - B (V ₂₈) =>167,67 - FL (V ₂₉) =>134,242 - B (V ₃₀) =>267,273 - FL (V ₃₁) =>287,247 - FL (V ₃₂)

Tabela 3 – Dados de entrada para os experimentos 1, 2 e 3

B (Bifurcação) FL (Final de Linha)

Para o quinto experimento, foram utilizados somente 3 minúcias de Fragmento e todas as minúcias de Candidata apresentadas nesta tabela.

As entradas para os dois primeiros experimentos como: lista de vértices, arestas, Ordem e Pai dos grafos G_F e G_C poderão ser conferidos no Anexo I. Por motivos de espaço as entradas para os outros experimentos não foram adicionadas no anexo. Para cada experimento serão mostrados os grafos de entrada, os mapeamentos (caso exista isomorfismo) e estatísticas sobre cada experimento.

Para cada experimento, os grafos construídos são apresentados em figuras onde cada vértice está disposto na imagem conforme sua geometria e está em destaque de acordo com o tipo de minúcia que o mesmo representa. Para as figuras apresentadas, os vértices em vermelho representam minúcias do tipo bifurcação e as em azul, os finais de linha.

6.1.1 Experimento nº 1: Fragmento e Candidata sendo representados por grafo Estrela de centro aleatório

Os grafos de entrada para este experimento podem ser visualizados na Figura 15.

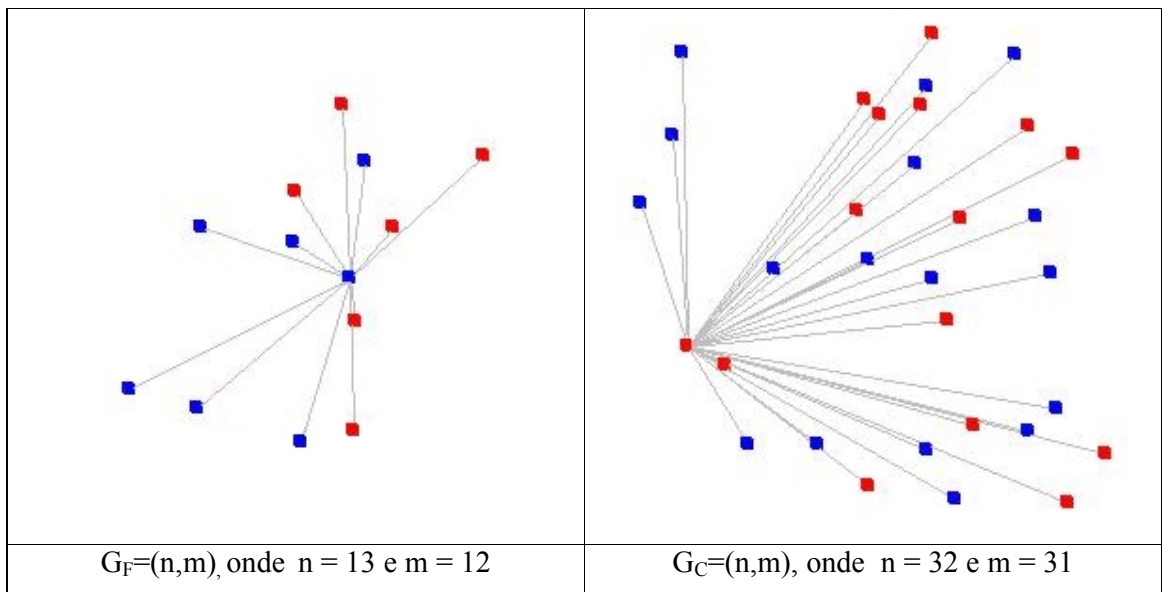


Figura 15 – Grafos de Entrada para o Experimento nº 1

6.1.1.1 Experimento nº 1a: sem controle no tamanho das arestas - discussão do Resultado

Para este experimento, apesar das imagens das impressões utilizadas para o teste representarem a mesma impressão digital e, comprovadamente para o conjunto de pontos X existir uma transformação T , tal que $T(X)$ que é um subconjunto de S , nenhum isomorfismo foi encontrado.

Este resultado foi negativo pelo fato de uma má escolha dos tipos de grafos para representar as impressões digitais. Como a construção dos grafos estrelas foi implementada aqui com vértice central variável, ocasionou que o centro de G_F fosse o vértice “7” e o de G_C fosse o vértice “19”. Desta forma, cada vez que uma transformação T era criada a partir da transformação da primeira aresta $a=ij$ de G_F com uma determinada aresta $b=kl$ de G_C , a busca prosseguia sobre os vértices w que constituem a listaOrdemF de G_F . Para cada w , buscou-se que vértice da aresta b , foi pré-mapeado para seu pai, conforme listaPai de G_F (algoritmo explanado na seção 4.4). Para este vértice, os seus vértices vizinhos y foram buscados para comparação geométrica com w . Nesta busca ou não haviam vértices vizinhos para realizar a comparação geométrica ou só foi encontrado 1 vértice vizinho que já estava pré-mapeado. Assim, nenhuma comparação geométrica foi realizada entre w e y .

Para este experimento os grafos escolhidos vieram a atrapalhar o resultado. O tempo estimado da busca foi de aproximadamente 31 milissegundos. Foram realizadas combinações a aresta $a=ij$ de G_F com todas as arestas de G_C por duas vezes (trocando as extremidades, pois os grafos não são orientados), mas pode-se perceber pelos tipos de minúcias de cada vértice que algumas combinações não puderam ser efetuadas totalizando 45 tentativas de combinação rejeitadas pelas análise das minúcias que seus os vértices representam, contra 17 que foram analisadas e realizadas pré-associações.

6.1.1.2 Experimento nº 1b: com controle no tamanho das arestas - discussão do Resultado

Visando diminuir o número de arestas de G_C a serem analisadas, este experimento realiza previamente uma análise no rótulo (que indica o seu tamanho) de cada aresta de G_C antes de analisar as minúcias que cada vértice representa e calcular o mapeamento geométrico desta com a aresta base, $a=ij$ de G_F . Para este experimento com controle no tamanho das

arestas esta aresta base é a aresta de menor tamanho de G_F e a $OrdemF$ é construída percorrendo no grafo esta aresta primeiro.

Para este teste as arestas com tamanho máximo de aproximadamente 41,25 foram analisadas. Pela lista de arestas de G_C , tem-se que de acordo com este critério somente a aresta $a_{19}=v_{19}v_{20}$, de tamanho aproximado: 22,36 pôde ser analisada. Porém, uma segunda condição, que analisa as minúcias que seus vértices representam, a descarta para qualquer construção da transformação T e possível pré-associação. Por esta condição, analisou-se que ambos seus dois vértices representam minúcias do tipo final de linha que foram comparados com as minúcias da aresta $a_3=v_7,v_3$ de G_F que representam uma bifurcação e uma minúcia do tipo final de linha respectivamente. Desta forma nenhuma transformação T pode ser construída ocasionando também que nenhuma comparação geométrica fosse realizada e nenhum isomorfismo encontrado. O tempo de busca foi de aproximadamente 16 milissegundos.

Neste caso, novamente o principal problema foi o grafo escolhido. O vértice central do grafo estrela, neste caso tenta simular o centro da impressão digital que é uma das minúcias mais complexas de se encontrar dentro da imagem. Se o centro fosse sempre o mesmo vértice, a análise da impressão digital por meio de grafos estrela seria uma boa abordagem para ser utilizada uma vez que é um grafo com menor grau se comparado à outros tipos de grafos que também serão apresentados aqui nos outros experimentos, como é o caso dos grafos bipartite e completo.

6.1.2 Experimento nº 2: Fragmento e Candidata sendo representadas por grafos Bipartite

Os grafos de entrada para este experimento podem ser visualizados na Figura 16.

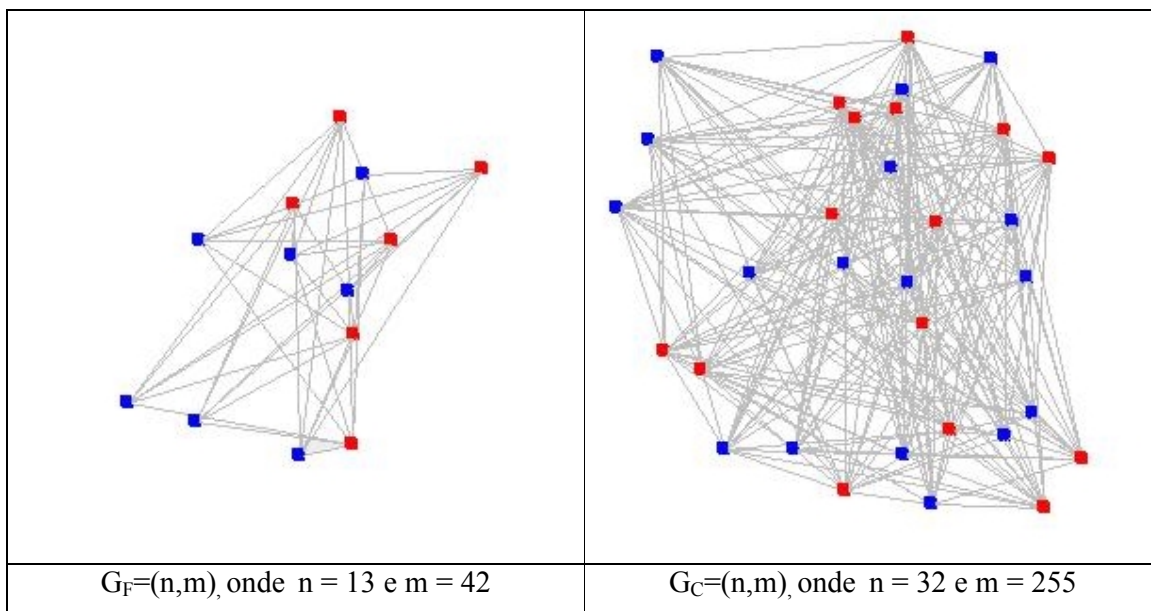


Figura 16 – Grafos de Entrada para o Experimento nº 2

6.1.2.1 Experimento nº 2a: sem controle no tamanho das arestas - discussão do Resultado

A busca localizou uma relação de isomorfismo. Esta relação de isomorfismo encontrada é apresentada na Tabela 4 e visualmente apresentada na Figura 17. Esta tabela mostra na primeira coluna o mapeamento encontrado de cada vértice de G_F para um determinado vértice G_C . Esta tabela apresenta também na segunda coluna a estatística contabilizada para que esta relação de isomorfismo fosse encontrada.

MAPEAMENTO	ESTATÍSTICA
VFrag: 1 VCand: 1	Nº Vértices de G_F : 13 Nº Arestas do G_F : 42 Nº Vértices de G_C : 32 Nº Arestas de G_C : 255 Nº Pré Associações Realizadas (minúcias): 5 Nº Pré Associações Não Realizadas (minúcias): 4 Nº de Comparações Geométricas Realizadas: 116 Tempo de execução: 32ms
VFrag: 5 VCand: 12	
VFrag: 6 VCand: 13	
VFrag: 7 VCand: 14	
VFrag: 8 VCand: 15	
VFrag: 9 VCand: 21	
VFrag: 10 VCand: 22	
VFrag: 13 VCand: 30	
VFrag: 2 VCand: 4	
VFrag: 3 VCand: 5	
VFrag: 4 VCand: 6	
VFrag: 11 VCand: 16	
VFrag: 12 VCand: 2	

Tabela 4 – Resultado principal do experimento nº 2a

A primeira coluna apresenta a relação dos identificadores dos vértices (VFrag) de G_F e os vértices (VCand) de G_C para os quais os vértices de G_F foram mapeados.

A segunda coluna apresenta as estatísticas para o experimento mostrando o tamanho e ordem G_F e G_C nas 4 (quatro) primeiras linhas, respectivamente. N° Pré Associações Realizadas, indica a quantidade de pré-associações da aresta a de G_F (aresta base) - transformações T - com arestas de G_C foram realizadas, sempre realizando o mapeamento de acordo com o tipo de minúcias que cada vértice representa. Por outro lado, N° Pré Associações Não Realizadas, indica quantas tentativas foram rejeitadas pelas minúcias dos vértices a serem mapeados não serem do mesmo tipo. O número de comparações geométricas para este experimento foi de 116 comparações e a relação de isomorfismo teve um tempo de execução de aproximadamente 32 milissegundos.

Um grafo bipartite é um grafo no qual as relações de vizinhança são estabelecidas somente para vértices que pertencem a duas diferentes classes. Para este exemplo, todos os vértices com minúcias do tipo bifurcação estão conectados com vértices que representam minúcias do tipo final de linha e vice-versa como na Figura 16. Se comparado ao grafo completo, este é um grafo com um menor número de arestas, porém a utilização do mesmo também tem desvantagem. Uma vez que em uma das impressões digitais (possivelmente na do Fragmento), podem ser encontradas somente minúcias de uma classe, ou Bifurcação ou Final de Linha e, se a mesma for representada por um grafo bipartite este não terá aresta, e nenhum mapeamento poderá ser feito para verificar a geometria.

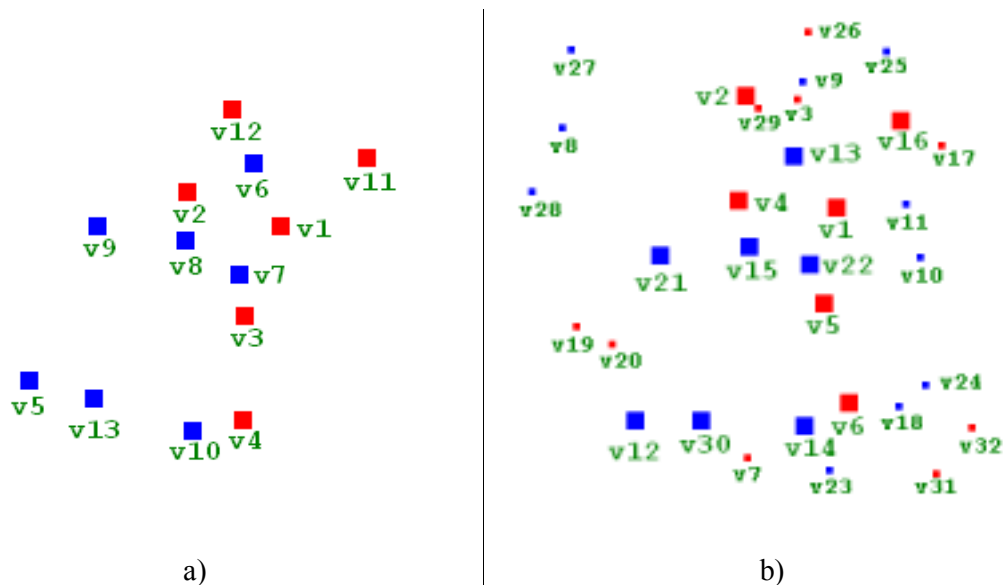


Figura 17 – Relação de isomorfismo entre os vértices de G_F e G_C . Vértices que tiveram pares encontrados aparecem em destaque

Após este isomorfismo ser encontrado, ainda existiam arestas de G_C que não tinham sido analisadas, por isso a busca continuou sobre mais 251 arestas. Realizando um total de 4113 comparações geométricas num tempo de execução de 563 milissegundos, mas nenhuma outra relação de isomorfismo foi encontrada.

6.1.2.2 Experimento nº 2b: com controle no tamanho das arestas - discussão do Resultado

De forma semelhante ao experimento nº 2a, a busca localizou uma relação de isomorfismo. A Tabela 5 apresenta os resultados alcançados, onde na primeira coluna é apresentado o mapeamento e na segunda as estatísticas.

MAPEAMENTO	ESTATÍSTICA
VFrag: 3 VCand: 5	
VFrag: 7 VCand: 14	Nº Vértices de G_F : 13
VFrag: 5 VCand: 12	Nº Arestas de G_F : 42
VFrag: 6 VCand: 13	Nº Vértices de G_C : 32
VFrag: 8 VCand: 15	Nº Arestas de G_C : 255
VFrag: 9 VCand: 21	Nº Arestas G_C - Analisadas (tamanho aresta): 6
VFrag: 10 VCand: 22	Nº Arestas G_C - Rejeitadas (tamanho aresta): 69
VFrag: 13 VCand: 30	Nº Pré Associações Realizadas (minúcias): 6
VFrag: 1 VCand: 1	Nº Pré Associações Não Realizadas (minúcias): 5
VFrag: 2 VCand: 4	Nº de Comparações Geométricas Realizadas: 137
VFrag: 4 VCand: 6	Tempo de execução: 62ms
VFrag: 11 VCand: 16	
VFrag: 12 VCand: 29	

Tabela 5 - Resultado principal do experimento nº 2b

A ordem em que este mapeamento foi encontrado, não é a mesma ordem do experimento 2a, porém se analisarmos os resultados vértice a vértice percebe-se que o mapeamento de um vértice de G_F para um vértice de G_C é o mesmo mapeamento encontrado no experimento anterior (Figura 17), com exceção do vértice 12 de G_F que para este experimento acabou sendo mapeado com o vértice 29 de G_C .

A ordem do mapeamento ser realizada é diferente pois na análise sem controle no tamanho das arestas, a aresta base de G_F é a primeira da lista de arestas. Já na análise com controle no tamanho das arestas é buscada pela aresta de menor tamanho de G_F para ser utilizada como base para localização de um par para a mesma em G_C .

Com relação ao diferente mapeamento para o vértice 12 de G_F , esta alteração deveu-se à ordem de busca no grafo que é de acordo com $OrdemF$ e também ao limiar geométrico utilizado para todos os experimentos apresentados aqui, cuja medida foi de 8.5.

O limiar geométrico foi utilizado para evitar que mapeamentos não sejam encontrados por erro ocasionado por arredondamentos realizados na criação da transformação geométrica. Para o experimento 2a de acordo com uma análise visual podemos certamente dizer que o vértice 12 foi mapeado erroneamente para o vértice 2 (assim o vértice 29 deixou de ser analisado), fato devido principalmente ao limiar geométrico utilizado que foi muito alto. Para os experimentos não foi tomada tanta cautela com relação ao limiar, porém uma boa opção de seu tamanho deve ser a menor distância geométrica entre os vértices do um grafo.

As estatísticas estão sendo demonstradas da mesma forma que no experimento 2a, com adição de duas outras estatísticas: N° Arestas G_C - Analisadas (tamanho aresta) e N° Arestas G_C - Rejeitadas (tamanho aresta). Estes dois itens nos fornecem quantas arestas de G_C foram analisadas (se o seu tamanho não ultrapassa o máximo tamanho previsto a partir do cálculo da proporcionalidade entre as arestas de G_C e G_F) e quantas foram rejeitadas e portanto não foram geradas uma transformação T entre elas e a aresta base, a (que é a menor aresta) de G_F .

Pelas estatísticas pode se visualizar que a análise de 69 arestas de G_C foram rejeitadas. Este número poderia ser reduzido e a resposta de isomorfismo positivo encontrada antes, caso a lista de arestas de G_C estivesse ordenada por ordem de tamanho, porém por outro lado, temos o custo em se ordenar esta lista de arestas e mais testes deveriam ser realizados para visualizar se compensa ordenar esta lista.

Após este isomorfismo ser encontrado, ainda tinham arestas da lista de G_C que não tinham sido analisadas, por isso a busca continuou sobre mais 181 arestas. Destas, 9 foram analisadas e 172 foram rejeitadas pelo seu tamanho. Um total de 142 comparações geométricas foram realizadas num tempo de execução de 109 milissegundos, porém sem nenhum sucesso, onde nenhuma outra relação de isomorfismo foi encontrada além da primeira já demonstrada na tabela.

6.1.3 Experimento nº 3: Fragmento e Candidata sendo representadas por grafos Completos

Os grafos de entrada para este experimento podem ser visualizados na Figura 18.

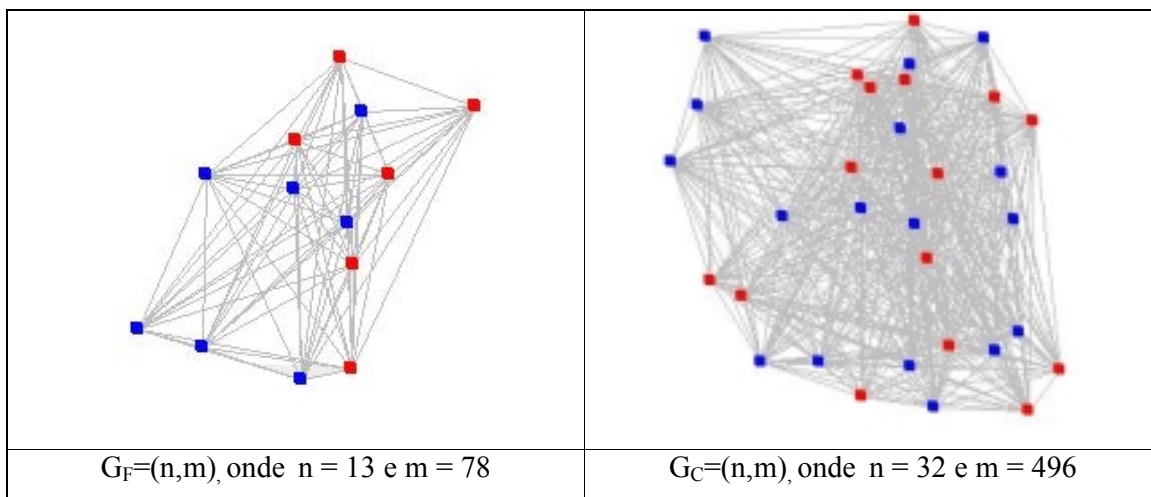


Figura 18 – Grafos de Entrada para o Experimento nº 3

6.1.3.1 Experimento nº 3a: sem controle no tamanho das arestas - discussão do Resultado

A busca localizou uma relação de isomorfismo. Esta relação de isomorfismo encontrada é apresentada na tabela Tabela 6. De forma semelhante aos experimentos anteriores, esta tabela nos mostra o mapeamento e as estatísticas para este isomorfismo.

MAPEAMENTO	ESTATÍSTICA
VFrag: 1 VCand: 1	Nº Vértices de G_F : 13 Nº Arestas de G_F : 78 Nº Vértices de G_C : 32 Nº Arestas de G_C : 496 Nº Pré Associações Realizadas (minúcias): 5 Nº Pré Associações Não Realizadas (minúcias): 0 Nº de Comparações Geométricas Realizadas: 211 Tempo de execução: 31ms
VFrag: 2 VCand: 4	
VFrag: 3 VCand: 5	
VFrag: 4 VCand: 6	
VFrag: 5 VCand: 12	
VFrag: 6 VCand: 13	
VFrag: 7 VCand: 14	
VFrag: 8 VCand: 15	
VFrag: 9 VCand: 21	
VFrag: 10 VCand: 22	
VFrag: 11 VCand: 16	
VFrag: 12 VCand: 2	
VFrag: 13 VCand: 30	

Tabela 6 - Resultado principal do experimento nº 2a

Semelhantemente aos outros experimentos realizados anteriormente com as mesmas minúcias detectadas, ainda que os grafos e a ordem de busca e procura sejam diferentes o mesmo mapeamento foi encontrado.

Após este isomorfismo ser encontrado, a busca prosseguiu por mais 494 arestas de G_C , com 205 pré-associações na tentativa de se buscar outra relação de isomorfismo. 6223

comparações geométricas foram realizadas num tempo de execução de 672 milissegundos. No entanto, nenhuma outra relação de isomorfismo foi encontrada.

6.1.3.2 Experimento nº 3b: com controle no tamanho das arestas - discussão do Resultado

Novamente, a busca localizou uma relação de isomorfismo. Esta relação de isomorfismo encontrada juntamente com as estatísticas para o experimento é apresentada na tabela abaixo (Tabela 7).

MAPEAMENTO	ESTATÍSTICA
VFrag: 3 VCand: 5	
VFrag: 7 VCand: 14	
VFrag: 1 VCand: 1	Nº Vértices do Frag: 13
VFrag: 2 VCand: 4	Nº Arestas do Frag: 78
VFrag: 4 VCand: 6	Nº Vértices da Cand: 32
VFrag: 5 VCand: 12	Nº Arestas da Cand: 496
VFrag: 6 VCand: 13	Nº Arestas Cand - Analisadas (tam): 9
VFrag: 8 VCand: 15	Nº Arestas Cand - Rejeitadas (tam): 118
VFrag: 9 VCand: 21	Nº Pré Associações Realizadas (minúcias): 6
VFrag: 10 VCand: 22	Nº Pré Associações Não Realizadas (minúcias): 11
VFrag: 11 VCand: 16	Nº de Comparações Geométricas Realizadas: 253
VFrag: 12 VCand: 29	Tempo de execução: 62ms
VFrag: 13 VCand: 30	

Tabela 7 - Resultado principal do experimento nº 2b

Semelhantemente aos outros experimentos realizados anteriormente com as mesmas minúcias detectadas, ainda que os grafos e a ordem de busca e procura sejam diferentes o mesmo mapeamento foi encontrado, com exceção do vértice 12 de G_F que foi mapeado para o vértice 29 de G_C pelos mesmos motivos explanados no experimento 2b.

Para encontrar esta relação de isomorfismo foram realizadas 6 pré-associações (construção de mapeamentos T), que implica em um número maior que o experimento anterior, porém este resultado pode ser melhorado se a lista de arestas de G_C for previamente ordenada pelo tamanho das arestas.

Após este isomorfismo ser encontrado, prosseguiu-se com uma nova busca sobre 370 arestas de G_C onde 17 delas possuíam o tamanho adequado para uma possível pré-associação. Destas 17, somente 8 combinações de vértices possuíam minúcias do mesmo tipo e puderam ser realmente analisadas juntamente com a aresta a de G_F (aresta base) e construir uma transformação T . Foram realizadas 268 comparações geométricas num tempo de execução de 140 milissegundos, porém nenhuma outra relação de isomorfismo foi encontrada.

6.1.4 Experimento nº 4: Fragmento representada por Grafo Estrela e Candidata representado por Grafo Completo

Os grafos de entrada para este experimento que representam as impressões digitais estão apresentados na Figura 19.

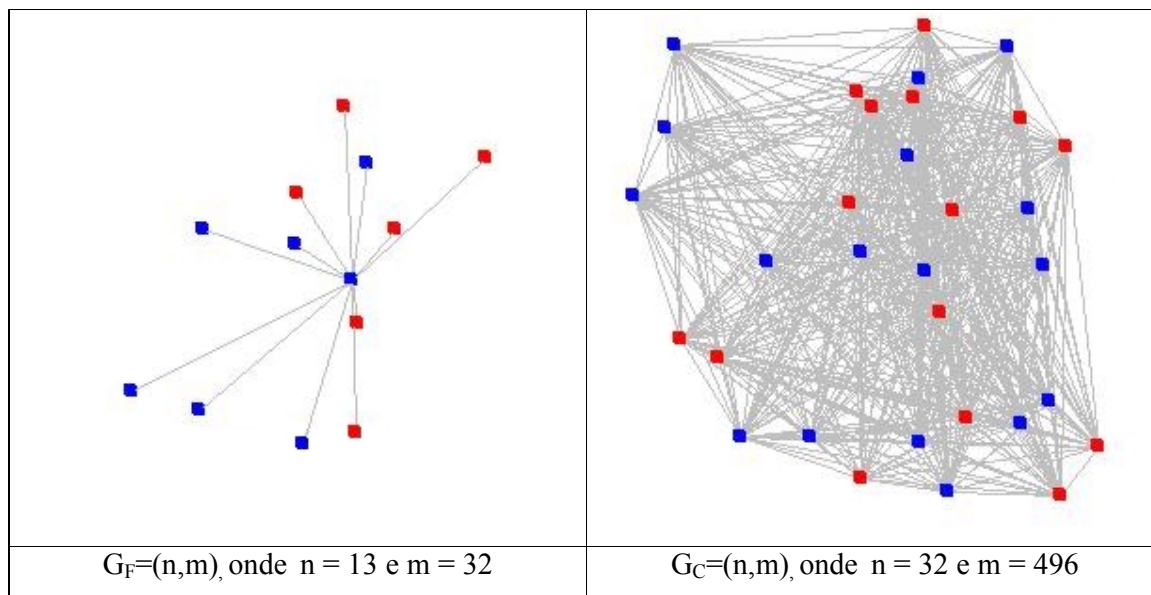


Figura 19 – Grafos de Entrada para o Experimento nº 4

Este experimento tem um diferencial dos demais mostrados, pois não visou representar a impressão Fragmento e Candidata pelo mesmo tipo de grafo. Na tentativa de se realizar um menor número de comparações para encontrar pelo menos uma solução, a impressão Fragmento foi representada por um grafo Estrela e a impressão Candidata por um grafo Completo. A medida do limiar geométrico para a comparação utilizado foi 8 (menor distância entre os dois vértices mais próximos de G_C que são os vértices v_2 e v_{29}).

Uma vez que os tipos de grafos escolhidos para este experimento contém as características de grafos necessárias para auxiliar na busca de uma solução (discutidas na seção 4.5), foram encontradas relações de isomorfismo tanto no experimento sem como no experimento com controle no tamanho das arestas.

Para cada experimento são apresentados os mapeamentos e as estatísticas encontrados estão apresentados na Tabela 8. Ao final desta seção são feitos comentários sobre os resultados alcançados.

Experimento 4a – Sem controle no tamanho das arestas

MAPEAMENTO	ESTATÍSTICA
VFrag: 7 VCand: 14	Nº Vértices do Frag: 13
VFrag: 1 VCand: 1	Nº Arestas do Frag: 12
VFrag: 2 VCand: 4	Nº Vértices da Cand: 32
VFrag: 3 VCand: 5	Nº Arestas da Cand: 496
VFrag: 4 VCand: 6	Nº Arestas Cand - Analisadas (tam): 13
VFrag: 5 VCand: 12	Nº Arestas Cand - Rejeitadas (tam): 1 (não considerado)
VFrag: 6 VCand: 13	Nº Pré Associações Realizadas (minúcias): 7
VFrag: 8 VCand: 15	Nº Pré Associações Não Realizadas (minúcias): 19
VFrag: 9 VCand: 21	Nº de Comparações Geométricas Realizadas: 296
VFrag: 10 VCand: 22	
VFrag: 11 VCand: 16	Total Nº Arestas Cand - Analisadas (tam): 496
VFrag: 12 VCand: 29	Total Nº Arestas Cand - Rejeitadas (tam): 1 (não considerado)
VFrag: 13 VCand: 30	Total Nº Pré Associações Realizadas (minúcias): 255
	Total Nº Pré Associações Não Realizadas (minúcias): 737
	Total Nº de Comparações Geométricas Realizadas: 7765
	Total Tempo de execução: 465ms

Experimento 4b – Com controle no tamanho das arestas

MAPEAMENTO	ESTATÍSTICA
VFrag: 7 VCand: 14	Nº Vértices do Frag: 13
VFrag: 3 VCand: 5	Nº Arestas do Frag: 12
VFrag: 1 VCand: 1	Nº Vértices da Cand: 32
VFrag: 2 VCand: 4	Nº Arestas da Cand: 496
VFrag: 4 VCand: 6	Nº Arestas Cand - Analisadas (tam): 27
VFrag: 5 VCand: 12	Nº Arestas Cand - Rejeitadas (tam): 100
VFrag: 6 VCand: 13	Nº Pré Associações Realizadas (minúcias): 16
VFrag: 8 VCand: 15	Nº Pré Associações Não Realizadas (minúcias): 38
VFrag: 9 VCand: 21	Nº de Comparações Geométricas Realizadas: 553
VFrag: 10 VCand: 22	
VFrag: 11 VCand: 16	Total Nº Arestas Cand - Analisadas (tam): 68
VFrag: 12 VCand: 29	Total Nº Arestas Cand - Rejeitadas (tam): 428
VFrag: 13 VCand: 30	Total Nº Pré Associações Realizadas (minúcias): 39
	Total Nº Pré Associações Não Realizadas (minúcias): 97
	Total Nº de Comparações Geométricas Realizadas: 1276
	Total Tempo de execução: 219ms

Tabela 8 - Resultados principais do experimento nº 4

A relação de isomorfismo encontrada para estes dois experimentos é a mesma encontrada para os experimentos de número 2b e 3b e apresentada com a rotulação dos vértices na Figura 20.

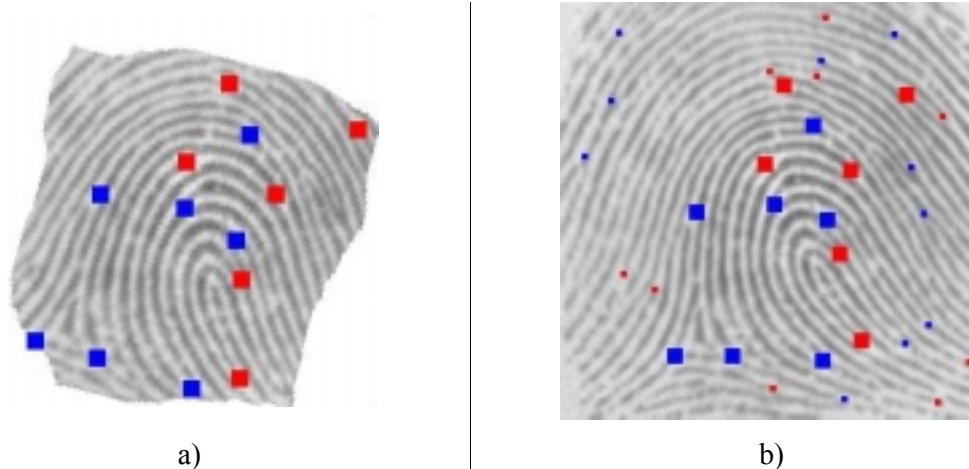


Figura 20 – Relação de isomorfismo entre os vértices de G_F e G_C . Vértices que tiveram pares encontrados aparecem em destaque

De acordo com a tabela de resultados dos experimentos para encontrar as relações de isomorfismo não houve diferença no tempo de execução para o experimento 4a e 4b, houve ainda um maior numero de arestas analisadas “Nº Arestas Cand – Analisadas (tam)” para o experimento 4b. Este maior número deveu-se principalmente que a lista de arestas de G_C não estava ordenada por tamanho que também por sua vez não esteve de acordo com OrdemF.

Apesar dos dois tempos de execução não serem diferentes para este caso e de um maior número de análise de arestas até que a relação de isomorfismo fosse encontrada, para o experimento considerando o tamanho das arestas que faz uma prévia restrição no número de arestas a serem utilizadas na comparação já reduziu em mais de 80% o número de arestas analisadas até todas as possíveis relações de isomorfismo fossem procuradas. Isso implicou que um segundo número, “Nº Pré Associações Realizadas (minúcias)”, também diminuísse. Este teste verifica para cada aresta se os vértices (minúcias) podem ser previamente mapeados de acordo com o seu tipo.

Outro número que também indica que a consideração no tamanho das arestas proposta neste trabalho é uma forma de otimização para a comparação das impressões digitais utilizando o isomorfismo de grafos pode ser comprovada pelo número total de “Comparações Geométricas Realizadas” que reduziu de 7765 no experimento sem consideração no tamanho das arestas para 1276 no experimento com consideração no tamanho das arestas.

6.1.5 Experimento nº 5: Fragmento representada por Grafo Estrela (com poucos vértices) e Candidata representada por Grafo Completo

Os grafos de entrada para este experimento que representam as impressões digitais estão apresentados na Figura 21.

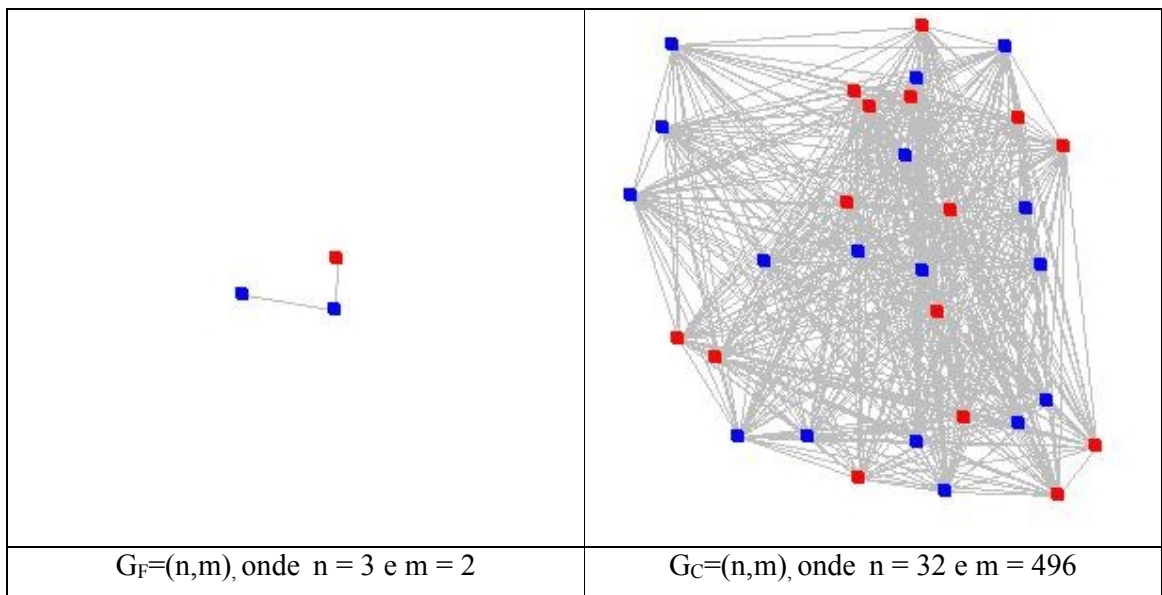


Figura 21 – Grafos de Entrada para o Experimento nº 5

Este experimento tem um outro diferencial dos demais mostrados, pois também não visou representar a impressão Fragmento e Candidata pelo mesmo tipo de grafo. Se forma semelhante ao experimento anterior, para este experimento a impressão Fragmento também foi representada por um grafo Estrela e a impressão Candidata por um grafo Completo.

O diferencial para este experimento é o número de minúcias encontradas na impressão fragmento ser somente de 3, ocasionando que G_F , tenha somente 3 vértices. A idéia deste experimento é mostrar que se o número de vértices de G_F é pequeno, podemos encontrar diversos mapeamentos, encontrando algumas vezes, falsos positivos.

Para este experimento foram encontradas 5 relações de isomorfismo que estão apresentadas na Tabela 9. Três destes mapeamentos podem ser visualizados na Figura 22 –. Esta figura apresenta em a) as três minúcias detectadas na impressão Fragmento, em b), c) e d) são apresentados os vértices que foram mapeados na impressão candidata para as três minúcias da impressão Fragmento. Estes são os mapeamentos 1, 2 e 5 respectivamente. Os vértices que foram mapeados aparecem destacados em tamanho maior.

Experimento 5 – G_F com $n = 3$				
MAPEAMENTO 1	MAPEAMENTO 2	MAPEAMENTO 3	MAPEAMENTO 4	MAPEAMENTO 5
$V_F: 2 \quad V_C: 15$	$V_F: 2 \quad V_C: 23$	$V_F: 2 \quad V_C: 10$	$V_F: 2 \quad V_C: 14$	$V_F: 2 \quad V_C: 28$
$V_F: 1 \quad V_C: 4$	$V_F: 1 \quad V_C: 6$	$V_F: 1 \quad V_C: 16$	$V_F: 1 \quad V_C: 16$	$V_F: 1 \quad V_C: 19$
$V_F: 3 \quad V_C: 21$	$V_F: 3 \quad V_C: 30$	$V_F: 3 \quad V_C: 21$	$V_F: 3 \quad V_C: 27$	$V_F: 3 \quad V_C: 13$

Tabela 9 – Mapeamentos encontrados para o experimento 5

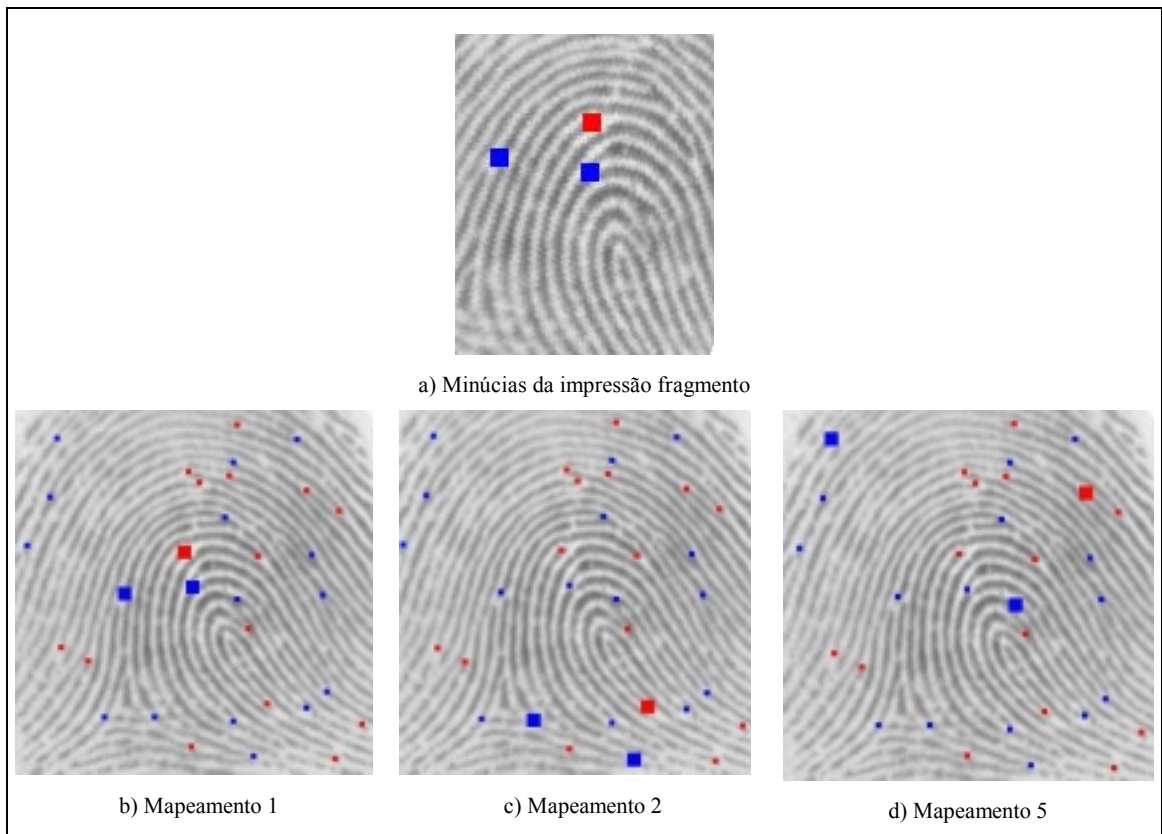


Figura 22 – Visualização de alguns dos mapeamentos encontrados para o experimento n° 5

7. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho teve por objetivo estudar e apresentar uma metodologia de comparação de impressões digitais baseada em isomorfismo de grafos. A partir dos pontos (minúcias) pré detectados na impressão digital foi definida uma metodologia para a comparação das impressões digitais que baseou-se na construção de grafos a partir destes pontos que vieram a ser os vértices dos grafos e depois definir relações de vizinhança entre os mesmos variando-se os tipos de grafos escolhidos.

Para escolher os tipos de grafos foram definidos critérios para esta escolha, pois a comparação é realizada através do cálculo do subgrafo da árvore geradora da impressão fragmento. Assim, o grafo de G_F deve possuir no mínimo as arestas que estão em G_C se as impressões digitais analisadas representarem a mesma impressão digital.

Analisado o problema, definimos uma abordagem adicionada ao cômputo do isomorfismo para que fossem realizadas um menor número de comparações. Esta abordagem – que analisa o tamanho das arestas de G_C – nos mostrou ser eficiente uma vez que ela não rejeitou arestas que não eram para serem rejeitadas e também reduziu o número de comparações (como esperado) que por sua vez reduziu o tempo de execução do cálculo do isomorfismo.

7.1 Trabalhos futuros

Para este trabalho os pontos foram detectados de forma manual a partir de imagens de impressão digital disponibilizadas na internet, assim a fase de processamento de imagens que é responsável por extrair os pontos não foi contemplada neste trabalho, sendo interessante que um algoritmo de processamento de imagens seja implementado para realizar a localização automática das minúcias.

Se um algoritmo eficiente para buscar as minúcias na impressão digital for implementado, poderá ser buscado por outras minúcias presentes na impressão digital, como o centro da mesma e não somente os finais de linha e bifurcação utilizados neste trabalho. Se o centro da impressão digital puder ser encontrado então poderão ser realizados mais testes representando G_F e G_C por grafos estrela para, comprovadamente provar sua eficiência.

Realizar testes com um maior número de impressões digitais, criando grafos sintéticos para buscar o isomorfismo simulando um banco de dados, no qual uma impressão digital pode ser comparada com N outras impressões e verificar as estatísticas.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [ARI2005] ARIYAEINIA, Alladin. *What is biometrics?* Disponível em:
http://www.fub.it/cost275/pages/home_biometrics/index.htm. Acesso em:
25 Jul. 2005
- [ATL2006] **Atlas de Grafos**. Disponível em:
http://www.fintec.edu.br/peter/relat04/atlas1/atlas_p1.htm#_Toc89180032
Acesso em: 30 jun. 2006.
- [BOL1997] BOLLE, Ruud et al. *An Identity Authentication System Using Fingerprints*.
In: Proceedings of the IEEE, v. 85, n. 9, set. 1997, p. 1365-1388. Disponível
em: <http://citeseer.nj.nec.com/jain97identity.html>. Acesso em: 4 jun. 2005.
- [BOL1979] BOLLOBÁS, Bela. **Graph Theory: An Introductory Course**. 1. ed. New
York: Springer-Verlag, 1979.
- [BOL1998] BOLLOBÁS, Béla. **Graduate Texts in Mathematics: Modern Graph
Theory**. Springer, 1ª Ed., 1998.
- [COR1970] CORNEIL, D. G., GOTLIEB, C. C., **An Efficient Algorithm for Graph
Isomorphism**. Journal of the ACM. Vol. 17, Issue 1. Pages: 51-64. Publisher:
ACM Press New York, NY, USA
- [COS2001] COSTA, Silvia Maria Farani. **Classificação e Verificação de Impressões
Digitais**. São Paulo: Escola Politécnica da Universidade de São Paulo, 2001
(Dissertação de Mestrado em Engenharia Elétrica)
- [DUA2004] DUARTE, Ângelo, TEDESCO, Claudia, COUTO, Daniel L.,
**Desenvolvimento de algoritmos para análise de imagens de impressões
digitais rotacionadas utilizando grafos**. In: CienteFico, Ano IV, v. I.
Salvador, janeiro-junho 2004 (Faculdade Ruy Barbosa). Disponível em:

<http://www.cientefico.frb.br/Impressa/Info/I.1.Duarte,A.Desenvolvimento%20de%20Algoritmos.pdf>. Acesso em: 30 jun. 2005

- [FRA2002] FRANKE, Katrin; KÖPPEN, Mario; RUIZ-DEL-SOLAR, Javier. *Soft-Biometrics: Soft-Computing for Biometric-Applications*. In: *International Journal of Fuzzy Systems*, v. 4, n. 2, p. 665-672, 2002.
- [ISE1986] ISENNOR, D. K., ZAKY, S. G. **Fingerprint identification using graph matching**. In: *Pattern Recognition*, v 19. n. 2, p. 113-122, ago de 1986.
- [GAG2005] GAGNON, Michel, **Conceitos de Grafos**. Disponível em: <http://www.inf.ufpr.br/~andre/Disciplinas/BSc/CI065/michel/Intro/intro.html#Iso> Acesso em: 24 ago. 2005.
- [GRE2005] GREVE, Fabiola. **Teoria dos Grafos e Algoritmos**. Disponível em: <http://twiki.im.ufba.br/bin/view/MAT156/AulaGrafos12>. Acesso em: 24 ago. 2005.
- [JAI2001] JAIN, Anil; PAKANTI, Sharath. *Automated Fingerprint Identification and Imaging Systems*. In: *Advances in Fingerprint Technology*, 2 ed. Science, 2001.
- [JAI2002] JAIN, Anil K., PANKANTI, Sharath, PRABHAKAR, Salil. *On the individuality of fingerprints*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, New York, v. 24, n. 8, p-1010-1025, 2002.
- [JAI2003] JAIN, Anil. K. et al. *Handbook of Fingerprint Recognition - Extract*. 1. ed. New York: Springer, 2003. Disponível em: <http://bias.csr.unibo.it/maltoni/handbook> Acesso em: 30 jun. 2005.
- [MAT1993] MATOS JÚNIOR, Álvaro C., **Especificação, implementação e validação de um método para comparação de impressões digitais utilizando grafos**

valorados. Curitiba, Centro Federal de Educação Tecnológica do Paraná, 1993 (Dissertação de Mestrado em Ciências).

[OLI2005] OLIVEIRA, M. O., GREVE, F. G., **Um novo algoritmo de refinamento para testes de isomorfismo em Grafos.** Departamento de Ciência da Computação. Universidade Federal da Bahia. Disponível em:
<http://twiki.im.ufba.br/pub/Gaudi/Publicacoes/sbc-reic-2005.pdf>. Acesso em:
24 ago. 2005.

[TEC2005] **TECNOLOGIA Arid.** Antheus Tecnologia: Análise e Reconhecimento de Impressões Digitais. Disponível em: <http://www.arid.com.br>. Acesso em: 11 jun. 2005.

ANEXO A – ENTRADAS PARA O EXPERIMENTO Nº 1

Fragmento	* Lista de Vértices		* Lista de Arestas	
	V1 (231.0,195.0) --> FL V2 (179.0,176.0) --> FL V3 (211.0,245.0) --> FL V4 (210.0,303.0) --> FL V5 (91.0,281.0) --> B V6 (216.0,160.0) --> B V7 (208.0,222.0) --> B V8 (178.0,203.0) --> B V9 (129.0,195.0) --> B V10 (182.0,309.0) --> B V11 (279.0,157.0) --> FL V12 (204.0,130.0) --> FL V13 (127.0,291.0) --> B		A1{v7v1} ** 35.47 A2{v7v2} ** 54.38 A3{v7v3} ** 23.19 A4{v7v4} ** 81.02 A5{v7v5} ** 131.03 A6{v7v6} ** 62.51	A7{v7v8} ** 35.51 A8{v7v9} ** 83.49 A9{v7v10} ** 90.8 A10{v7v11} ** 96.26 A11{v7v12} ** 92.09 A12{v7v13} ** 106.4
Candidata	* Lista de Vértices		* Lista de Arestas	
	V1 (210.0,122.0) --> FL V2 (159.0,59.0) --> FL V3 (189.0,62.0) --> FL V4 (155.0,118.0) --> FL V5 (203.0,176.0) --> FL V6 (217.0,232.0) --> FL V7 (161.0,264.0) --> FL V8 (57.0,78.0) --> B V9 (192.0,52.0) --> B V10 (258.0,151.0) --> B V11 (250.0,121.0) --> B V12 (97.0,242.0) --> B V13 (186.0,93.0) --> B V14 (195.0,154.0) --> B V15 (161.0,144.0) --> B V16 (246.0,73.0) --> FL V17 (270.0,88.0) --> FL V18 (246.0,235.0) --> B V19 (65.0,190.0) --> FL V20 (85.0,200.0) --> FL V21 (111.0,149.0) --> B V22 (192.0,245.0) --> B V23 (207.0,271.0) --> B V24 (261.0,223.0) --> B V25 (239.0,35.0) --> B V26 (195.0,24.0) --> FL V27 (62.0,34.0) --> B V28 (40.0,114.0) --> B V29 (167.0,67.0) --> FL V30 (134.0,242.0) --> B V31 (267.0,273.0) --> FL V32 (287.0,247.0) --> FL		A1{v19v1} ** 160.15 A2{v19v2} ** 161.24 A3{v19v3} ** 178.21 A4{v19v4} ** 115.26 A5{v19v5} ** 138.71 A6{v19v6} ** 157.7 A7{v19v7} ** 121.21 A8{v19v8} ** 112.29 A9{v19v9} ** 187.54 A10{v19v10} ** 196.9 A11{v19v11} ** 197.45 A12{v19v12} ** 61.06 A13{v19v13} ** 155.08 A14{v19v14} ** 134.89 A15{v19v15} ** 106.45 A16{v19v16} ** 215.52	A17{v19v17} ** 228.97 A18{v19v18} ** 186.51 A19{v19v20} ** 22.36 A20{v19v21} ** 61.62 A21{v19v22} ** 138.4 A22{v19v23} ** 163.48 A23{v19v24} ** 198.76 A24{v19v25} ** 233.03 A25{v19v26} ** 210.85 A26{v19v27} ** 156.03 A27{v19v28} ** 80.01 A28{v19v29} ** 159.79 A29{v19v30} ** 86.4 A30{v19v31} ** 218.39 A31{v19v32} ** 229.2
Fragmento	Lista Ordem e Pai - Experimento 1a		Lista Ordem e Pai - Experimento 1b	
	Vért: 7 Pai: 0 Vért: 1 Pai: 7 Vért: 2 Pai: 7 Vért: 3 Pai: 7 Vért: 4 Pai: 7 Vért: 5 Pai: 7 Vért: 6 Pai: 7	Vért: 8 Pai: 7 Vért: 9 Pai: 7 Vért: 10 Pai: 7 Vért: 11 Pai: 7 Vért: 12 Pai: 7 Vért: 13 Pai: 7	Vért: 7 Pai: 0 Vért: 1 Pai: 7 Vért: 2 Pai: 7 Vért: 3 Pai: 7 Vért: 4 Pai: 7 Vért: 5 Pai: 7 Vért: 6 Pai: 7	Vért: 8 Pai: 7 Vért: 9 Pai: 7 Vért: 10 Pai: 7 Vért: 11 Pai: 7 Vért: 12 Pai: 7 Vért: 13 Pai: 7

ANEXO B – ENTRADAS PARA O EXPERIMENTO N° 2

Fragmento	* Lista de Vértices		* Lista de Arestas			
	V1 (231.0,195.0) --> FL		A1{v1v5} ** 164.3	A15{v3v5} ** 125.28	A29{v5v11} ** 225.21	
	V2 (179.0,176.0) --> FL		A2{v1v6} ** 38.08	A16{v3v6} ** 85.15	A30{v5v12} ** 188.6	
	V3 (211.0,245.0) --> FL		A3{v1v7} ** 35.47	A17{v3v7} ** 23.19	A31{v6v11} ** 63.07	
	V4 (210.0,303.0) --> FL		A4{v1v8} ** 53.6	A18{v3v8} ** 53.41	A32{v6v12} ** 32.31	
	V5 (91.0,281.0) --> B		A5{v1v9} ** 102.0	A19{v3v9} ** 96.04	A33{v7v11} ** 96.26	
	V6 (216.0,160.0) --> B		A6{v1v10} ** 124.08	A20{v3v10} ** 70.26	A34{v7v12} ** 92.09	
	V7 (208.0,222.0) --> B		A7{v1v13} ** 141.53	A21{v3v13} ** 95.77	A35{v8v11} ** 110.98	
	V8 (178.0,203.0) --> B		A8{v2v5} ** 137.0	A22{v4v5} ** 121.02	A36{v8v12} ** 77.49	
	V9 (129.0,195.0) --> B		A9{v2v6} ** 40.31	A23{v4v6} ** 143.13	A37{v9v11} ** 154.74	
	V10 (182.0,309.0) --> B		A10{v2v7} ** 54.38	A24{v4v7} ** 81.02	A38{v9v12} ** 99.25	
	V11 (279.0,157.0) --> FL		A11{v2v8} ** 27.02	A25{v4v8} ** 105.0	A39{v10v11} ** 180.31	
	V12 (204.0,130.0) --> FL		A12{v2v9} ** 53.49	A26{v4v9} ** 135.0	A40{v10v12} ** 180.35	
	V13 (127.0,291.0) --> B		A13{v2v10} ** 133.03	A27{v4v10} ** 28.64	A41{v11v13} ** 202.63	
			A14{v2v13} ** 126.21	A28{v4v13} ** 83.86	A42{v12v13} ** 178.47	

Candidata	* Lista de Vértices		* Lista de Arestas	
	V1 (210.0,122.0) --> FL	A1{v1v8} ** 159.2	A86{v6v8} ** 222.07	A171{v14v20} ** 119.23
	V2 (159.0,59.0) --> FL	A2{v1v9} ** 72.28	A87{v6v9} ** 181.73	A172{v14v26} ** 130.0
	V3 (189.0,62.0) --> FL	A3{v1v10} ** 56.08	A88{v6v10} ** 90.79	A173{v14v29} ** 91.39
	V4 (155.0,118.0) --> FL	A4{v1v11} ** 40.01	A89{v6v11} ** 115.8	A174{v14v31} ** 139.09
	V5 (203.0,176.0) --> FL	A5{v1v12} ** 164.83	A90{v6v12} ** 120.42	A175{v14v32} ** 130.82
	V6 (217.0,232.0) --> FL	A6{v1v13} ** 37.64	A91{v6v13} ** 142.41	A176{v15v16} ** 110.75
	V7 (161.0,264.0) --> FL	A7{v1v14} ** 35.34	A92{v6v14} ** 81.04	A177{v15v17} ** 122.54
	V8 (57.0,78.0) --> B	A8{v1v15} ** 53.71	A93{v6v15} ** 104.31	A178{v15v19} ** 106.45
	V9 (192.0,52.0) --> B	A9{v1v18} ** 118.6	A94{v6v18} ** 29.15	A179{v15v20} ** 94.4
	V10 (258.0,151.0) --> B	A10{v1v21} ** 102.62	A95{v6v21} ** 134.63	A180{v15v26} ** 124.72
	V11 (250.0,121.0) --> B	A11{v1v22} ** 124.31	A96{v6v22} ** 28.18	A181{v15v29} ** 77.23
	V12 (97.0,242.0) --> B	A12{v1v23} ** 149.03	A97{v6v23} ** 40.26	A182{v15v31} ** 166.96
	V13 (186.0,93.0) --> B	A13{v1v24} ** 113.15	A98{v6v24} ** 44.91	A183{v15v32} ** 162.74
	V14 (195.0,154.0) --> B	A14{v1v25} ** 91.71	A99{v6v25} ** 198.22	A184{v16v18} ** 162.0
	V15 (161.0,144.0) --> B	A15{v1v27} ** 172.19	A100{v6v27} ** 251.45	A185{v16v21} ** 154.92
	V16 (246.0,73.0) --> FL	A16{v1v28} ** 170.19	A101{v6v28} ** 212.73	A186{v16v22} ** 180.28
	V17 (270.0,88.0) --> FL	A17{v1v30} ** 142.04	A102{v6v30} ** 83.6	A187{v16v23} ** 201.8
	V18 (246.0,235.0) --> B	A18{v2v8} ** 103.75	A103{v7v8} ** 213.1	A188{v16v24} ** 150.75
	V19 (65.0,190.0) --> FL	A19{v2v9} ** 33.73	A104{v7v9} ** 214.25	A189{v16v25} ** 38.64
	V20 (85.0,200.0) --> FL	A20{v2v10} ** 135.15	A105{v7v10} ** 148.92	A190{v16v27} ** 188.09
	V21 (111.0,149.0) --> B	A21{v2v11} ** 110.11	A106{v7v11} ** 168.43	A191{v16v28} ** 210.04
	V22 (192.0,245.0) --> B	A22{v2v12} ** 193.22	A107{v7v12} ** 67.68	A192{v16v30} ** 202.74
	V23 (207.0,271.0) --> B	A23{v2v13} ** 43.42	A108{v7v13} ** 172.82	A193{v17v18} ** 148.95
	V24 (261.0,223.0) --> B	A24{v2v14} ** 101.59	A109{v7v14} ** 115.13	A194{v17v21} ** 170.3
	V25 (239.0,35.0) --> B	A25{v2v15} ** 85.02	A110{v7v15} ** 120.0	A195{v17v22} ** 175.31
	V26 (195.0,24.0) --> FL	A26{v2v18} ** 196.33	A111{v7v18} ** 89.81	A196{v17v23} ** 193.54
	V27 (62.0,34.0) --> B	A27{v2v21} ** 102.0	A112{v7v21} ** 125.4	A197{v17v24} ** 135.3
	V28 (40.0,114.0) --> B	A28{v2v22} ** 188.9	A113{v7v22} ** 36.36	A198{v17v25} ** 61.4
	V29 (167.0,67.0) --> FL	A29{v2v23} ** 217.37	A114{v7v23} ** 46.53	A199{v17v27} ** 214.9
	V30 (134.0,242.0) --> B	A30{v2v24} ** 193.13	A115{v7v24} ** 108.08	A200{v17v28} ** 231.46
	V31 (267.0,273.0) --> FL	A31{v2v25} ** 83.52	A116{v7v25} ** 241.92	A201{v17v30} ** 205.46
	V32 (287.0,247.0) --> FL	A32{v2v27} ** 100.17	A117{v7v27} ** 250.4	A202{v18v19} ** 186.51
		A33{v2v28} ** 131.1	A118{v7v28} ** 192.72	A203{v18v20} ** 164.76
		A34{v2v30} ** 184.7	A119{v7v30} ** 34.83	A204{v18v26} ** 217.08
		A35{v3v8} ** 132.97	A120{v8v16} ** 189.07	A205{v18v29} ** 185.65
		A36{v3v9} ** 10.44	A121{v8v17} ** 213.23	A206{v18v31} ** 43.42
		A37{v3v10} ** 112.61	A122{v8v19} ** 112.29	A207{v18v32} ** 42.72
		A38{v3v11} ** 84.86	A123{v8v20} ** 125.17	A208{v19v21} ** 61.62
		A39{v3v12} ** 202.15	A124{v8v26} ** 148.19	A209{v19v22} ** 138.4
		A40{v3v13} ** 31.14	A125{v8v29} ** 110.55	A210{v19v23} ** 163.48
		A41{v3v14} ** 92.2	A126{v8v31} ** 286.57	A211{v19v24} ** 198.76
		A42{v3v15} ** 86.65	A127{v8v32} ** 285.41	A212{v19v25} ** 233.03
		A43{v3v18} ** 182.15	A128{v9v16} ** 57.94	A213{v19v27} ** 156.03
		A44{v3v21} ** 116.85	A129{v9v17} ** 85.91	A214{v19v28} ** 80.01
		A45{v3v22} ** 183.02	A130{v9v19} ** 187.54	A215{v19v30} ** 86.4
		A46{v3v23} ** 209.77	A131{v9v20} ** 182.63	A216{v20v21} ** 57.25
		A47{v3v24} ** 176.37	A132{v9v26} ** 28.16	A217{v20v22} ** 116.08
		A48{v3v25} ** 56.82	A133{v9v29} ** 29.15	A218{v20v23} ** 141.16
		A49{v3v27} ** 130.05	A134{v9v31} ** 233.38	A219{v20v24} ** 177.5
		A50{v3v28} ** 157.81	A135{v9v32} ** 216.91	A220{v20v25} ** 225.7
		A51{v3v30} ** 188.22	A136{v10v16} ** 78.92	A221{v20v27} ** 167.59
		A52{v4v8} ** 105.85	A137{v10v17} ** 64.13	A222{v20v28} ** 97.06
		A53{v4v9} ** 75.66	A138{v10v19} ** 196.9	A223{v20v30} ** 64.54
		A54{v4v10} ** 108.16	A139{v10v20} ** 179.81	A224{v21v26} ** 150.6
		A55{v4v11} ** 95.05	A140{v10v26} ** 141.77	A225{v21v29} ** 99.3
		A56{v4v12} ** 136.89	A141{v10v29} ** 123.84	A226{v21v31} ** 199.28
		A57{v4v13} ** 39.82	A142{v10v31} ** 122.33	A227{v21v32} ** 201.44
		A58{v4v14} ** 53.81	A143{v10v32} ** 100.28	A228{v22v26} ** 221.02
		A59{v4v15} ** 26.68	A144{v11v16} ** 48.17	A229{v22v29} ** 179.75
		A60{v4v18} ** 148.22	A145{v11v17} ** 38.59	A230{v22v31} ** 80.06
		A61{v4v21} ** 53.82	A146{v11v19} ** 197.45	A231{v22v32} ** 95.02
		A62{v4v22} ** 132.28	A147{v11v20} ** 182.94	A232{v23v26} ** 247.29
		A63{v4v23} ** 161.6	A148{v11v26} ** 111.51	A233{v23v29} ** 207.88
		A64{v4v24} ** 149.2	A149{v11v29} ** 99.02	A234{v23v31} ** 60.03
		A65{v4v25} ** 118.09	A150{v11v31} ** 152.95	A235{v23v32} ** 83.52
		A66{v4v27} ** 125.32	A151{v11v32} ** 131.32	A236{v24v26} ** 209.66
		A67{v4v28} ** 115.07	A152{v12v16} ** 225.3	A237{v24v29} ** 182.13
		A68{v4v30} ** 125.77	A153{v12v17} ** 231.61	A238{v24v31} ** 50.36
		A69{v5v8} ** 175.84	A154{v12v19} ** 61.06	A239{v24v32} ** 35.38
		A70{v5v9} ** 124.49	A155{v12v20} ** 43.68	A240{v25v26} ** 45.35
		A71{v5v10} ** 60.42	A156{v12v26} ** 239.01	A241{v25v29} ** 78.79
		A72{v5v11} ** 72.35	A157{v12v29} ** 188.48	A242{v25v31} ** 239.64
		A73{v5v12} ** 124.87	A158{v12v31} ** 172.8	A243{v25v32} ** 217.37
		A74{v5v13} ** 84.72	A159{v12v32} ** 190.07	A244{v26v27} ** 133.38
		A75{v5v14} ** 23.41	A160{v13v16} ** 63.25	A245{v26v28} ** 179.23
		A76{v5v15} ** 52.8	A161{v13v17} ** 84.15	A246{v26v30} ** 226.37
		A77{v5v18} ** 73.01	A162{v13v19} ** 155.08	A247{v27v29} ** 110.06
		A78{v5v21} ** 95.88	A163{v13v20} ** 147.14	A248{v27v31} ** 314.87
		A79{v5v22} ** 69.87	A164{v13v26} ** 69.58	A249{v27v32} ** 309.83
		A80{v5v23} ** 95.08	A165{v13v29} ** 32.2	A250{v28v29} ** 135.42
		A81{v5v24} ** 74.65	A166{v13v31} ** 197.39	A251{v28v31} ** 277.15
		A82{v5v25} ** 145.52	A167{v13v32} ** 184.17	A252{v28v32} ** 280.53
		A83{v5v27} ** 200.11	A168{v14v16} ** 95.72	A253{v29v30} ** 178.08
		A84{v5v28} ** 174.39	A169{v14v17} ** 99.9	A254{v30v31} ** 136.57
		A85{v5v30} ** 95.48	A170{v14v19} ** 134.89	A255{v30v32} ** 153.08

Fragmento	Lista Ordem e Pai - Experimento 2a		Lista Ordem e Pai - Experimento 2b	
	Vért: 1 Pai: 0 Vért: 5 Pai: 1 Vért: 6 Pai: 1 Vért: 7 Pai: 1 Vért: 8 Pai: 1 Vért: 9 Pai: 1 Vért: 10 Pai: 1	Vért: 13 Pai: 1 Vért: 2 Pai: 5 Vért: 3 Pai: 5 Vért: 4 Pai: 5 Vért: 11 Pai: 5 Vért: 12 Pai: 5	Vért: 3 Pai: 0 Vért: 5 Pai: 3 Vért: 6 Pai: 3 Vért: 7 Pai: 3 Vért: 8 Pai: 3 Vért: 9 Pai: 3 Vért: 10 Pai: 3	Vért: 13 Pai: 3 Vért: 1 Pai: 5 Vért: 2 Pai: 5 Vért: 4 Pai: 5 Vért: 11 Pai: 5 Vért: 12 Pai: 5